

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра автоматизації та управління в технічних системах**

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Програмне забезпечення
інформаційно-комунікаційних систем»
спеціальності 121 «Інженерія програмного забезпечення»
на тему: «Мобільний застосунок системи управління вивезенням
відсортованого сміття»**

Виконав (-ла):

студент (-ка) IV курсу, групи ІТ-61

Венчик Євгеній Станіславович _____

Керівник:

декан ФІОТ, д.т.н., проф.

Теленик Сергій Федорович _____

Рецензент:

асистент кафедри ТК, к. т. н.

Сергеев Данило Сергійович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент (-ка) _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення інформаційно-комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Венчик Євгеній Станіславович

1. Тема проєкту «Мобільний застосунок системи управління вивезенням відсортованого сміття», керівник проєкту Теленик Сергій Федорович, доктор технічних наук, професор, затверджені наказом по університету від «7» травня 2020 р. №1081-с
2. Термін подання студентом проєкту 09 червня 2020 року
3. Вихідні дані до проєкту: система управління вивезенням відсортованого сміття, датчик виміру заповненості контейнера зі сміттям, мобільний застосунок, який буде маршрут через заповнені контейнери.
4. Зміст пояснювальної записки: визначити мету дипломного проєкту, проаналізувати існуючі рішення, розробити систему управління вивезенням відсортованого сміття, розробити мобільний застосунок системи управління вивезенням відсортованого сміття, розробити керівництво користувача

мобільним застосунком, дослідити систему управління вивезенням відсортованого сміття.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): діаграма варіантів використання мобільного застосунку системи, схема макета системи, діаграма послідовності системи, схема структурна системи.

6. Дата видачі завдання 4 березня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд існуючих рішень	13.03.2020	Виконано
2	Розробка структурної схеми системи	27.03.2020	Виконано
3	Розробка макету контейнера з датчиком заповненості	12.04.2020	Виконано
4	Розробка мобільного застосунку	15.05.2020	Виконано
5	Налаштовування сервісів Firebase	18.05.2020	Виконано
6	Дослідження системи	25.05.2020	Виконано
7	Розробка керівництва користувача мобільного застосунку	26.05.2020	Виконано
8	Оформлення пояснювальної записки	07.06.2020	Виконано

Студент

Євгеній ВЕНЧИК

Керівник

Сергій ТЕЛЕНИК

АНОТАЦІЯ

Венчик Є. С. Мобільний застосунок системи управління вивезенням відсортованого сміття. КПІ ім. І. Сікорського, Київ, 2020.

Проект містить 69 с. тексту, 25 рисунків, 4 таблиці, посилання на 24 літературні джерела, 1 додаток та 4 конструкторських документи.

Ключові слова: вивезення сміття, відсортоване сміття, мобільний застосунок, оптимізація, управління, Android, сервіс Firebase.

Об'єктом дослідження є процес вивезення сміття міськими службами та підприємствами.

Мета розробки полягає в скороченні витрат на вивезення сміття, в тому числі й відсортованого, шляхом визначення маршрутів через точки вивозу в яких знаходяться лише заповнені контейнери.

У дипломному проєкті розроблено систему управління вивезенням відсортованого сміття та її прототип. Система включає такі компоненти: мобільний застосунок, датчик виміру заповненості контейнеру зі сміттям разом з мікроконтролером, та центральний обробник даних разом з базою даних. Мобільний застосунок – головний компонент системи, оскільки прокладає маршрути через точки вивозу із заповненими контейнерами зі сміттям.

Отримані результати розробки можуть бути використані при автоматизації процесів вивезення сміття.

SUMMARY

Venchyk Y. S. Mobile application of sorted garbage collection management system. Igor Sikorsky KPI, Kyiv, 2020.

The project contains 69 pages. text, 25 figures, 4 tables, links to 24 literary sources, 1 annex and 4 design documents.

Keywords: garbage collection, sorted garbage, mobile application, optimization, management, Android, Firebase service.

The object of development is the process of garbage collection by local services and enterprises.

The purpose of the development – cost reduction of garbage collection process, including sorted garbage, by building routes through garbage collection spots that include only filled garbage containers.

The graduation project developed garbage collection management system and its prototype. The system includes following components: mobile application, filling measurement sensor of garbage container together with microcontroller, and central data processor together with database. The mobile application is the main component of the system, as it builds routes through garbage collection spots with filled garbage containers only.

The obtained results can be useful in automating garbage collections process.

№ рядка	Формат	Позначення	Найменування	Кіл. листів	№ прим.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	ІТ61.070БАК.004 ПЗ	Пояснювальна записка	69		
6	A3	ІТ61.070БАК.004 Д1	Система управління	1		
7			вивезенням			
8			відсортованого сміття.			
9			Схема структурна.			
10	A3	ІТ61.070БАК.004 Д2	Мобільний застосунок	1		
11			системи управління			
12			вивезенням			
13			відсортованого сміття.			
14			Діаграма варіантів			
15			використання.			
16	A3	ІТ61.070БАК.004 Д3	Система управління	1		
17			вивезенням			
18			відсортованого сміття.			
19			Діаграма послідовності.			
20	A3	ІТ61.070БАК.004 Д3	Система управління	1		
21			вивезенням			
22			відсортованого сміття.			
23			Схема макета системи.			
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
					ІТ61.070БАК.004 ТП	
		Прізвище	Підпис	Дат		
Розроб.		Венчик Є.С..			Мобільний застосунок системи управління вивезенням відсортованого сміття. Відомість дипломного проекту	
Перевірив.						
Н. кон.						
Затв.						
					Лім.	Лист
					Т	1
						1
					КПІ ім. Ігоря Сікорського Каф. АУТС гр. ІТ-61	

**Пояснювальна записка
до дипломного проєкту
на тему: «Мобільний застосунок системи
управління вивезенням відсортованого сміття»**

Київ – 2020 року

ЗМІСТ

ВСТУП.....	6
1 ОГЛЯД ІСНУЮЧИХ МОБІЛЬНИХ ЗАСТОСУНКІВ ВИВОЗУ СМІТТЯ...	9
1.1 Мобільний застосунок водія сміттєвоза	9
1.2 Sensoneo Driver Navigation	10
1.3 RG-Soft вивіз сміття.....	10
1.4 Результат огляду існуючих рішень	11
Висновки до розділу	12
2 РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ ВИВЕЗЕННЯМ ВІДСОРТОВАНОГО СМІТТЯ	14
2.1 Розробка структурної схеми системи управління вивезенням відсортованого сміття	14
2.2 Розробка функціональної моделі системи управління вивезенням відсортованого сміття	16
2.2.1 Особливості методології функціонального моделювання IDEF0.....	16
2.2.2 Призначення функціональної моделі системи управління вивезенням відсортованого сміття	17
2.2.3 Розробка функціональної моделі системи управління вивезенням відсортованого сміття	18
2.3 Вибір датчику вимірювання заповненості контейнера.....	20
2.4 Вибір центрального обробника даних	21
2.4.1 Сервіс Firebase.....	21
2.5 Вибір бази даних	22
2.5.1 Розробка структури бази даних системи управління вивезенням відсортованого сміття	22

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		2

2.6	Автоматичне робоче місце адміністратора	24
2.7	Автоматичне робоче місце водія сміттєвоза.....	24
	Висновки до розділу	24
3	РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ СИСТЕМИ УПРАВЛІННЯ ВИВЕЗЕННЯМ ВІДСОРТОВАНОГО СМІТТЯ	26
3.1	Призначення мобільного застосунку	26
3.2	Аналіз предметної області мобільного застосунку	26
3.3	Розробка функціональної моделі мобільного застосунку	27
3.4	Структура програмних компонентів мобільного застосунку.....	27
3.5	Вибір операційної системи для мобільного застосунку	30
3.5.1	Операційна система Android.....	30
3.5.2	Операційна система IOS.....	32
3.5.3	Результат аналізу операційних систем	32
3.6	Вибір мов програмування для розробки мобільного застосунку	34
3.6.1	Мова програмування Java	34
3.6.2	Мова програмування Kotlin	35
3.6.3	Результат огляду мов програмування для розробки мобільного застосунку.....	36
3.7	Вибір картографічних сервісів та сервісів побудови маршрутів для розробки мобільного застосунку	37
3.7.1	Сервіс Google Maps.....	37
3.7.2	Сервіси Here	38
3.7.3	Сервіс OpenStreetMaps	38
3.7.4	Результат аналізу картографічних сервісів та сервісів побудови маршрутів для розробки мобільного застосунку	39
3.8	Середовище розробки мобільного застосунку.....	40

3.9 Використані методи та алгоритми при розробці мобільного застосунку.	41
3.9.1 Метод виявлення точок, що входять в круг між двома точками	41
3.9.2 Алгоритм сортування точок за наближеністю до заданої.....	42
3.10 Структура вихідного коду мобільного застосунку.....	43
3.11 Структура графічних ресурсів мобільного застосунку.....	47
3.12 Діаграми класів екранів мобільного застосунку.....	47
3.13 Діаграма послідовності роботи компонентів системи управління вивезенням відсортованого сміття	51
Висновки до розділу	52
4 РОЗРОБКА КЕРІВНИЦТВА КОРИСТУВАЧА МОБІЛЬНОГО ЗАСТОСУНКУ	53
4.1 Визначення системних вимог пристрою для запуску мобільного застосунку	53
4.2 Керівництво користувача	53
Висновки до розділу	58
5 ДОСЛІДЖЕННЯ СИСТЕМИ УПРАВЛІННЯ ВИВЕЗЕННЯМ ВІДСОРТОВАНОГО СМІТТЯ	59
5.1 Мета дослідження системи управління вивезенням відсортованого сміття.....	59
5.2 Огляд досліджуваної системи управління вивезенням відсортованого сміття.....	59
5.3 Дослідження побудови маршрутів через точки вивозу заповнених контейнерів зі сміттям	61
Висновки до розділу	65
ВИСНОВКИ.....	66

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 67

Додаток А 69

					ІТ61.070БАК.004 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дат		

ВСТУП

Сьогодні в більшості міст України накопичуються десятки тонн твердих побутових відходів, які необхідно вчасно вивозити. Здавалося б, така звичайна та непомітна робота, але тільки завдяки вчасному вивозу твердих побутових відходів ми маємо змогу насолоджуватись чистотою наших вулиць та красою навколишньої природи. Проте, не налагоджений своєчасний вивіз сміття відразу позначається на чистоті вулиць та навколишньої природи.

На даний момент ефективна підтримка бізнес-процесів, у тому числі вивозу сміття, здійснюється за допомогою автоматизації ІТ-послугами, як на рівні між підприємствами, так і всередині підприємств [1]. Сучасні комунальні підприємства, які здійснюють вивезення сміття, не мають грошей та ресурсів на створення та підтримку власних ІТ-підрозділів, які могли б займатися автоматизацією власних бізнес-процесів. Виходом є створення окремої ІТ-інфраструктури, тобто сукупність інформаційних технологій, мереж, ресурсів та пристроїв кінцевого користувача, яка буде окремо займатися розробкою та підтримкою автоматизації певних бізнес процесів, і надавати послуги підприємствам.

У 2018 році кількість користувачів смартфонів в Україні складала 85% серед активних користувачів інтернету [2]. Кожного року їх кількість зростає. Причиною цього є широкий вибір смартфонів в різних цінових категоріях. Навіть у найдешевших смартфонах зараз є вбудовані: GPS модуль, який дозволяє визначати точне місцезнаходження смартфона, Wi-Fi та мобільний інтернет, які дозволяють отримувати доступ до мережі Інтернет. Розвиненість можливостей смартфона дозволяє розробникам створювати застосунки для розв'язання певних задач.

На сьогоднішній день міські служби або підприємства, що здійснюють вивіз твердих побутових відходів в Україні не здатні забезпечити своєчасне та оперативне вивезення сміття. Всі підприємства в Україні, які займаються цією діяльністю, створюють фіксований графік та маршрути вивозу твердих

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		6

побутових відходів. Такий підхід не враховує ситуацій, коли машина для вивозу сміття приїжджає до точки вивозу, а більшість контейнерів є напівпорожніми. Машина просто проїжджає маршрут напівпорожньою. За такі напівпусті виїзди користувачам послуг вивозу твердих побутових відходів доводиться платити гроші. Причиною виникнення цієї проблеми є відсутність актуальних даних про заповненість контейнерів твердих побутових відходів та відсутність належних інструментів їх обробки у підприємств. Вивезення відсортованого сміття фіксованим графіком стає не вигідним для здійснення підприємствами, оскільки контейнери з різними типами сміття наповнюються з різною швидкістю та потребують вивезення окремо для кожного типу сміття.

Одним зі шляхів розв'язання цієї проблеми є створення системи управління вивезенням відсортованого сміття, яка буде складатися із мобільного застосунку, центрального сховища даних та датчика виміру заповненості контейнера.

Об'єктом дослідження даного дипломного проєкту є процес вивезення сміття міськими службами та підприємствами.

Предметом даного дипломного проєкту є інструмент управління вивезенням сміття.

Завдання дипломного проєкту полягає у створенні мобільного застосунку, який буде частиною системи управління вивезенням сміття.

Мета полягає в скороченні витрат на вивезення сміття, в тому числі й відсортованого, шляхом визначення маршрутів через точки вивозу, в яких знаходяться лише заповнені контейнери. Зменшивши відвідування точок вивозу з незаповненими контейнерами, зменшується собівартість вивозу сміття.

Засобом досягнення поставленої мети є мобільний застосунок системи управління вивезенням відсортованого сміття, який допомагає розв'язати проблему з побудовою оптимальних маршрутів вивозу сміття. Маршрут має будуватися через заповнені контейнери, виключивши ситуації з відвідуванням незаповнених контейнерів. Це зменшить витрати підприємствам на здійснення вивезення сміття. Для досягнення цього застосунок має отримувати актуальні

					ІТ61.070БАК.004 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дат		

дані про заповненість точок вивозу твердих побутових відходів. Критерії заповненості смітників мають вказуватися користувачем. Застосунок повинен будувати маршрути як для смітників зі звичайними твердими побутовими відходами, так і для смітників з відсортованим сміттям.

Тому тема бакалаврського проєкту, яка присвячена оптимізація вивозу сміття за допомогою мобільного застосунку є актуальною.

Дипломний проєкт складається з наступних розділів: вступ, огляд існуючих мобільних застосунків, розробка структури системи управління вивезенням відсортованого сміття, розробка мобільного застосунку системи управління вивезенням відсортованого сміття, керівництво користувача мобільного застосунку, дослідження системи управління вивезенням відсортованого сміття, висновок та перелік використаних джерел.

					ІТ61.070БАК.004 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дат		

1 ОГЛЯД ІСНУЮЧИХ МОБІЛЬНИХ ЗАСТОСУНКІВ ВИВОЗУ СМІТТЯ

Існує багато рішень пов'язаних з управлінням вивезенням сміття. Провівши дослідження, знайдено декілька аналогічних мобільних застосунків, які є частинами систем, що здійснюють управління вивезенням сміття.

1.1 Мобільний застосунок водія сміттєвоза

Першим таким рішенням є Мобільний застосунок водія сміттєвоза [3], який є частиною програмно-апаратного комплексу «Керування відходами». Окрім мобільного застосунку в комплекс також входить застосунок для збору та управління даними від користувачів послуг та водіїв сміттєвозів, та апаратного датчика, який встановлений на машині, що здійснює вивіз сміття.

Головною функцією мобільного застосунку є карта, яка показує маршрут, по якому повинен проїхати сміттєвоз та положення самого сміттєвоза на карті. Положення сміттєвоза відстежується за допомогою апаратного датчика. Диспетчер, отримавши дані від клієнтів про заповненість обслуговуючих контейнерів, що обслуговуються, вносить їх до спеціального застосунку, що є частиною цього програмно-апаратного комплексу. Далі, за допомогою цього застосунку диспетчер створює маршрут вивозу сміття.

Серед інших функцій мобільний застосунок водія сміттєвоза має:

- відображення списку точок вивозу, які потрібно відвідати сміттєвозу;
- відображення денних задач та графіки їх виконання;
- перегляд детальної інформації про точки вивозу, які обслуговує оператор вивозу твердих побутових відходів;
- можливість створення звіту про зроблену роботу;
- онлайн чат з диспетчером.

Перевагою даного застосунку є його головна функція – інтерактивна карта, яка відображає маршрут, що спрощує водіям сміттєвозів виконувати їх роботу.

					ІТ61.070БАК.004 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дат		

Недоліком додатку є те, що маршрут задається диспетчером вручну на основі отриманих даних від користувачів послуги вивозу сміття, що не вирішує проблеми, поставленої в цьому дипломному проєкті. Також прокладений маршрут не адаптується до поточної позиції водія на карті.

Ціна мобільного застосунку – безкоштовно, але для його роботи потрібен програмно апаратний комплекс «Керування відходами».

1.2 Sensoneo Driver Navigation

Аналогічним рішенням є мобільний застосунок Sensoneo Driver Navigation [4]. Застосунок працює разом з програмою Smart Route Planning. Головною функцією застосунку є прокладення оптимального маршруту проїзду через контейнери, які задав попередньо диспетчер. Диспетчер задає контейнери, опираючись на інформацію про заповненість контейнерів, отриману від клієнтів. На карті відображається прокладений маршрут відносно поточного положення водія. Кінцевою точкою маршруту застосунок вибирає найближче сміттєзвалище. Окрім того застосунок надає водію альтернативні маршрути здійснення вивезення сміття.

Застосунок має такі додаткові функції:

- відстеження поточного місцезнаходження водія на карті;
- створення заявки про проблему з конкретним сміттєвим баком.

Недоліком застосунку є відсутність можливості отримувати актуальні дані про заповненість смітників та будувати маршрут лише через заповнені смітники.

Мобільний застосунок є безкоштовним, але для його повноцінної роботи потрібна програма Sensoneo Smart Route Planning.

1.3 RG-Soft вивіз сміття

Аналогічним рішенням є мобільний застосунок RG-Soft вивіз сміття [5], який розроблений на платформі 1С:Підприємство. Головною функцією

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		10

застосунку є завантаження маршрутних листів через платформу 1С:Підприємство для водіїв смітєвозів. Далі водій, під час здійснення вивозу сміття, може відмічати точки вивозу, які він вже відвідав. По закінченню маршруту водій сповіщає про його завершення та застосунок автоматично генерує звіт.

Застосунок має такі додаткові функції:

- відмічання кількості одиниць зібраного сміття з точки вивозу;
- створення фото звіту, якщо не вдається здійснити вивіз сміття з точки вивозу;
- відстеження поточного місцезнаходження водія.

Недоліком застосунку є відсутність карти, яка відображає маршрут водієві, показує його місцезнаходження на маршруті, відсутність можливості отримувати актуальні дані про заповненість смітників на точках вивозу та можливість будувати маршрут через заповнені смітники.

Мобільний застосунок є безкоштовним, але для його роботи потрібно налаштувати програмне забезпечення на платформі 1С:Підприємство.

1.4 Результат огляду існуючих рішень

Провівши огляд існуючих рішень, наведено порівняння функцій цих застосунків в таблиці 1.1.

Таблиця 1.1 – Порівняння функцій розглянутих рішень

Функції	Мобільний застосунок водія смітєвоза	Sensoneo driver navigation	RG-Soft вивіз сміття
Інтерактивна карта	+	+	—

Функції	Мобільний застосунок водія сміттєвоза	Sensoneo driver navigation	RG-Soft вивіз сміття
Задання точок вивозу зовнішньою програмою	+	+	+
Побудова маршруту через точки вивозу сміття застосунком	—	+	—
Місцезнаходження водія	+	+	+
Актуальні дані про заповненість смітників	—	—	—

Проаналізувавши існуючі рішення, зроблено висновок, що в Україні немає мобільного застосунку системи управління вивезенням відсортованого сміття, який зміг би вирішити поставлену проблему в даному дипломному проєкті. Застосунку, який зміг би будувати маршрут лише через заповнені точки вивозу сміття, зменшивши собівартість вивозу сміття. Жоден із застосунків не має доступу до актуальних даних про заповненість контейнерів.

Найбільший набір функцій присутній в мобільному застосунку Sensoneo driver navigation. Дані функції можуть бути використані для досягнення поставленої мети в даному дипломному проєкті.

Висновки до розділу

В результаті проведення огляду існуючих мобільних застосунків систем управління вивезенням відсортованого сміття визначено, що в Україні немає системи управління вивезенням відсортованого сміття, яка мала б актуальні дані про заповненість контейнерів.

Визначені компоненти розглянутих систем управління: мобільний застосунок, центральний обробник даних, база даних з даними про заповненість

контейнерів. Визначено такі функції для розроблюваного мобільного застосунку, розглянувши функції існуючих мобільних застосунків:

- отримання та перегляд актуальних даних про заповненість смітника;
- вибір точок вивозу сміття, через які потрібно будувати маршрут;
- встановлення критерію заповненості смітників;
- відображення місцезнаходження точок вивозу контейнерів на карті;
- відображення поточного місцезнаходження користувача на карті;
- адаптування маршруту до поточного місцезнаходження водія.

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		13

2 РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ ВИВЕЗЕННЯМ ВІДСОРТОВАНОГО СМІТТЯ

У розділі описана структурна та функціональна схема системи управління вивезенням відсортованого сміття. Визначені та описані компоненти системи управління вивезенням відсортованого сміття.

2.1 Розробка структурної схеми системи управління вивезенням відсортованого сміття

Враховуючи мету, поставлену в цьому дипломному проєкті та аналіз існуючих мобільні застосунків систем управління вивезенням відсортованого сміття, розроблена структурна схема системи, яка зображена на кресленнику IT61.070БАК.004 Д1.

Структура системи управління вивезенням відсортованого сміття складається з наступних компонентів: контейнер зі сміттям, датчик вимірювання заповненості контейнера та модуль зв'язку з Інтернетом, які розміщені на мікроконтролері, центральний обробник даних, база даних з даними про заповненість всіх контейнерів, автоматизоване робоче місце адміністратора та автоматизоване робоче місце водія.

Контейнер зі сміттям – компонент системи управління вивезенням відсортованого сміття, який представляє собою спеціалізовану ємність, яка використовується для збору всіх твердих побутових відходів або лише певних видів. Може мати довільний об'єм. Контейнери розташовується в точках вивозу твердих побутових відходів.

Мікроконтролер – компонент системи управління вивезенням відсортованого сміття, на якому розташовуються датчик вимірювання заповненості контейнера та модуль зв'язку з інтернетом. Цей компонент отримує актуальну інформацію про заповненість контейнера від датчика та відправляє її

					IT61.070БАК.004 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дат		

до центрального обробника даних через Інтернет. Сам мікроконтролер розташовується всередині контейнера зі сміттям.

Датчик – компонент системи управління вивезенням відсортованого сміття, який розташовується на мікроконтролері. Задача датчика вимірювати заповненість контейнеру зі сміттям та передавати ці дані мікроконтролеру.

Модуль зв'язку з інтернетом – компонент системи управління вивезенням відсортованого сміття, який розташовується на мікроконтролері та дозволяє йому відправляти дані через інтернет.

Центральний обробник даних – компонент системи управління вивезенням відсортованого сміття, який приймає або віддає інформацію про заповненість контейнерів зі сміттям. Взаємодіє з базою даних, яка зберігає інформацією про смітники напрямку. Автоматизовані робочі місця адміністратора та водія сміттєвоза взаємодіють з центральним обробником даних через мережу Інтернет. Задача центрального обробника даних отримувати актуальні дані про заповненість контейнерів та зберігати їх в базу даних. Потім надавати ці дані автоматизованому робочому місцю водія.

База даних – компонент системи управління вивезенням відсортованого сміття, яка зберігає та надає доступ до актуальних даних про заповненість контейнерів. База даних взаємодіє напрямку з центральним обробником даних.

Автоматизоване робоче місце адміністратора – компонент системи управління вивезенням відсортованого сміття, який забезпечує доступ адміністраторам системи до управління нею. Автоматизоване робоче місце адміністратора може бути у вигляді web-клієнта, який взаємодіє через інтернет з центральним обробником даних.

Автоматизоване робоче місце водія – компонент системи управління вивезенням відсортованого сміття, який відображає актуальні дані про заповненість контейнерів зі сміттям у точках вивозу та забезпечує побудову маршруту через заповнені контейнери певних типів. Автоматизоване робоче місце водія може бути у вигляді мобільного застосунку, який взаємодіє з центральним обробником даних через Інтернет.

					ІТ61.070БАК.004 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дат		

2.2 Розробка функціональної моделі системи управління вивезенням відсортованого сміття

Розробка функціональної моделі системи управління вивезенням відсортованого сміття здійснена за допомогою методології IDEF0 [6].

2.2.1 Особливості методології функціонального моделювання IDEF0

Методологією функціонального моделювання IDEF0 називають технологію, яка описує систему як сукупність пов'язаних між собою дій, або функцій. Ці функції розглядаються незалежно від об'єктів системи, що виконують їх. Це допомагає визначити чіткі аспекти призначення системи, абстрагуючись від її реалізації.

Найчастіше цю методологію використовують для дослідження та проектування систем на логічному рівні.

При побудові спочатку визначається призначення моделі, тобто для чого створюється модель. Далі визначаються границі моделі – ширина охоплення предметної області та глибина деталізації. Це є логічним продовженням призначення моделі.

Сама модель складається із діаграми, яка містить лише два позначення: блоки та стрілки. Блоками на діаграмі позначають дії або функції, що обробляють або переводять вхідні дані на результуючі дані. Оскільки модель являє собою набір ієрархічних функцій, що вкладені один в одного, спочатку потрібно визначити функцію, яка визначає систему в цілому, тобто контекстну функцію. Назви функцій вибираються з використанням дієслів, віддієслівних іменників так, щоб вони відображали сутність системи з точки зору, яку вибрали для моделювання. Будь-який блок може бути декомпозований на окремі блоки.

Опис блока повинен відображати результат своєї роботи – виходи, та об'єктів, які блок перетворює – входи. В моделі IDEF0 присутні позначення керування та механізми виконання. Під керуванням розуміють об'єкти, які

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		16

впливають на спосіб перетворення вхідних об'єктів на вихідні об'єкти. Під механізмом керування розуміють об'єкти, які безпосередньо виконують роботу перетворення вхідних даних на вихідні.

Опис блока проводиться за допомогою стрілок. Стрілки входу завжди направлені в ліву частину прямокутника, який позначає функціональний блок. Наявність вхідних стрілок на діаграмі не є необхідним, так як є можливість того, що деякі блоки нічого не перетворюють.

Стрілка керування завжди направлена у верхню частину прямокутника, який позначає функціональний блок.

Стрілка входу завжди виходить з правої частини прямокутника, який позначає функціональний блок. Кожен блок повинен мати мінімум один вихід. Дія, яка не виробляє ніяких чітко визначених дій не повинна моделюватися взагалі.

Стрілка механізму виконання завжди направлена в нижню частину прямокутника, який позначає функціональний блок. Дані стрілки можуть бути відсутні, у випадку, якщо виявиться, що вони не являються необхідними для досягнення цілі моделювання.

2.2.2 Призначення функціональної моделі системи управління вивезенням відсортованого сміття

Призначення функціональної моделі системи управління вивезенням відсортованого сміття є визначення функціональних принципів роботи системи, які потрібні для досягнення поставленої в цьому дипломному проєкті мети. Скоротити витрати на вивезення сміття, в тому числі й відсортованого, шляхом визначення маршрутів через точки вивозу в яких знаходяться лише заповнені контейнери.

					ІТ61.070БАК.004 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дат		

2.2.3 Розробка функціональної моделі системи управління вивезенням відсортованого сміття

Контекстна функція системи управління вивезенням відсортованого сміття описана на рисунку 2.1.

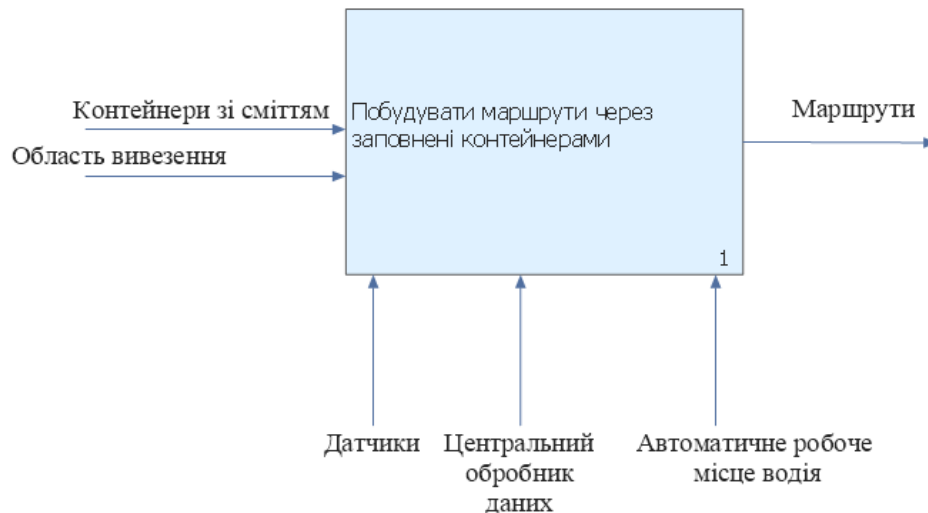


Рисунок 2.1 – Функціональна модель системи управління вивезенням відсортованого сміття

На вхід система управління вивезенням відсортованого сміття отримує контейнери зі сміттям. Це об'єкти, які мають такі властивості:

- заповненість;
- тип сміття;
- точка вивозу.

На виході система управління вивезенням відсортованого сміття надає об'єкт маршрути, що проходять через заповнені контейнери зі сміттям.

Здійснює система перетворення об'єктів контейнерів на об'єкти маршрутів за допомогою: датчиків, центрального обробника даних та автоматичного робочого місця водія.

Провівши декомпозицію контекстної функції, отримав детальний опис функціональної моделі системи управління вивезенням відсортованого сміття, яка зображена на рисунку 2.2.



Рисунок 2.2 – Декомпозиція першого рівня функціональної моделі системи управління вивезенням відсортованого сміття

Спочатку проводиться вимірювання заповненості контейнеру за допомогою датчику. Далі дані про вимірювання отримує центральний обробник даних. Центральний обробник даних зберігає ці дані у своєму сховищі. Далі центральний обробник даних надсилає автоматичному робочому місцю водія точки вивозу сміття. Автоматичне робоче місце водія фільтрує точки вивозу відповідно до заданої області вивезення заповненості. Далі автоматичне робоче місце водія будує маршрут через ці точки вивозу, які мають заповнені контейнери зі сміттям.

2.3 Вибір датчику вимірювання заповненості контейнера

Відповідно до розділу 2.1, задачею датчика є вимірювання заповненості контейнеру та відправлення цих даних на центральний обробник даних. Контролювати ці процеси повинен мікроконтролер, який підключений до датчика. Контролер має вирішувати такі задачі, як підключення до мережі Інтернет, отримання даних із датчика вимірювання заповненості та надсилання даних на відповідний центральний обробник даних.

У якості прототипу датчика з мікроконтролером в даному дипломному проєкті використовується мікроконтролер Arduino ESP8266 IOT Elecrow [7] з ультразвуковим датчиком відстані HC-SR04 [8]. Прототип зображено на рисунку 2.3.

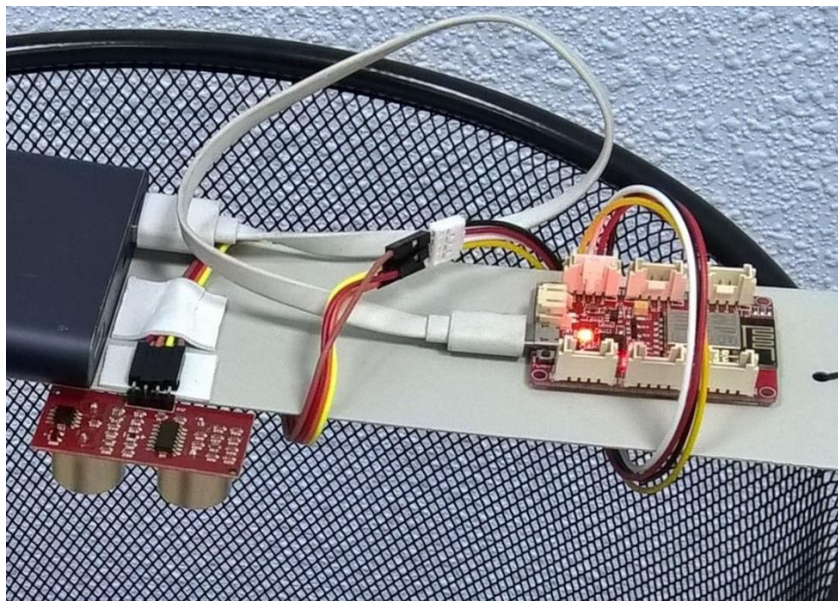


Рисунок 2.3 – Прототип датчику виміру заповненості контейнера та мікроконтролер

Ультразвуковий датчик працює за принципом ультразвукової ехолокації. У складі датчика є два елемента, один працює як випромінювач сигналу, інший як приймач сигналу. Випромінювач генерує сигнал, який відбивається від перешкод та потрапляє на приймач. Вимірюючи час за який сигнал пройде до перешкоди і назад можна визначити відстань.

Мікроконтролер Arduino ESP8266 IOT Elecrow має вбудований WI-FI модуль, який дозволяє приєднуватися до бездротової мережі інтернет. Також в мікроконтролері присутній micro-USB порт.

2.4 Вибір центрального обробника даних

Відповідно до пункту 2.1, центральний обробник даних містить в собі базу даних з інформацією про смітники та застосунок адміністратора. Щоб не витрачати час на розробку центрального обробника даних та розгортання самого серверу з базою даних, в якості готового рішення з базою даних при розробці прототипу системи управління вивезенням відсортованого сміття використаний сервіс Firebase [9] від Google.

2.4.1 Сервіс Firebase

Firebase – це набір готових back-end інструментів, таких як: база даних, API, віддалені конфігурації та багато інших, які вже розгорнені на серверах Google. Головною перевагою цього сервісу є Firebase SDK, яка дозволяє інтегрувати ці рішення для мобільних та проєкт застосунків.

Визначення інструментів, які вибрані для розробки, що є у Firebase, представлені в таблиці А.1

Окрім інструментів для розробки у Firebase представлено ще інструменти для відстеження роботи мобільних та проєкт застосунків:

- Crashlytics;
- Performance;
- Test Lab;
- App Distribution.

Інструменти для збору та аналізу аналітичних метрик, проведення А/В тестів, організацію відправки повідомлень клієнтам.

					ІТ61.070БАК.004 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дат		

Для початку роботи з даним сервісом потрібно перейти в консоль, яка знаходиться на сайті Firebase, зареєструватися там та створити проєкт, вказавши його назву.

2.5 Вибір бази даних

У якості бази даних при розробці прототипу системи управління вивезенням відсортованого сміття використане рішення Firebase Realtime Database, особливості якого були описані в таблиці 2.1. Перевагу було надано само цьому рішенню перед Firebase Cloud Store через наявність бібліотеки для передачі даних з Arduino ESP8266, та більш швидкої роботи з примітивними типами даних.

Структура бази даних представляє JSON документ, тобто об'єкт, який має символічний ключ, та значення, яке може бути числом, символічним рядком, датою, масивом або об'єктом, який складається із попередніх типів. В ієрархії бази даних Firebase Realtime Database на найвищому рівні розташований ключ з ідентифікатором бази даних у Firebase. Значенням ключа є об'єкт у якій розробник може додавати свої дані у вигляді пари ключ – значення.

2.5.1 Розробка структури бази даних системи управління вивезенням відсортованого сміття

При проєктуванні бази даних були визначені такі сутності, які мають зберігатися в ній:

- типи сміття, яке містить контейнер;
- координати точок вивозу контейнерів;
- заповненість контейнерів.

Структура бази даних системи управління вивезенням відсортованого сміття представлена на рисунку 2.4.

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		22

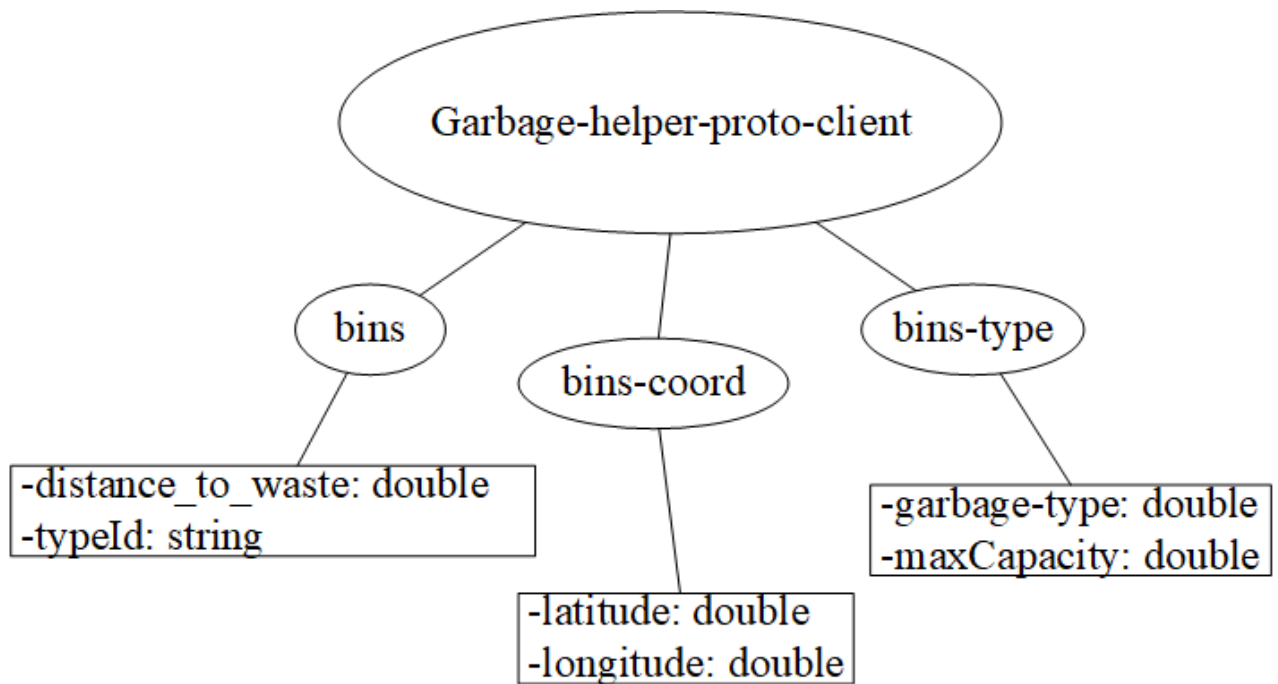


Рисунок 2.4 – Структура бази даних у Firebase Realtime Database у вигляді дерева

База даних складається з трьох ключів:

- bins;
- bins-coord;
- bins-type.

bins – масив об'єктів, які зберігають дані про заповненість та тип контейнеру та ключів цих об'єктів, які є ідентифікатором контейнера. Об'єкт має такі ключі:

- distance_to_waste – тип double, відображає відстань до сміття;
- typeId – тип string, ідентифікатор типу контейнера.

bins-coord – масив об'єктів, які зберігають дані про координати точок вивозу контейнерів та ключі, які є ідентифікатори відповідних контейнерів. Об'єкт має такі ключі:

- latitude – тип double, значення широти точки вивозу контейнера;
- longitude – тип double, значення довготи точки вивозу контейнера.

bins-type – масив об'єктів, які зберігають дані про тип контейнера та ключі, які є назвами цих типів. Об'єкт має такі ключі:

– garbage-type – тип double, ідентифікатор типу сміття, який міститься в контейнері;

– maxCapacity – тип double, максимальна довжина смітника.

Можливі такі ідентифікатори типів сміття: 1 – будь-яке сміття, 2 – скло та склобій, 3 – папір та картон, 4 – пластик, 5 – органічні відходи, 6 – акумуляторні батареї.

2.6 Автоматичне робоче місце адміністратора

У якості автоматичного робочого місця адміністратора системи управління вивезенням відсортованого сміття, в рамках створення прототипу цієї системи, вирішено використовувати Firebase консоль, оскільки розробка повноцінного автоматичного робочого місця адміністратора потребує багато часу. Firebase консоль має повний доступ до Firebase Realtime Database. Тому доцільно використовувати консоль як заміну застосунку адміністрування.

2.7 Автоматичне робоче місце водія сміттєвоза

Головною частиною системи управління вивезення відсортованого сміття є автоматичне робоче місце водія сміттєвоза. У якості реалізації автоматичного робочого місця водія вибраний мобільний застосунок, який здійснює побудову маршрутів через заповнені контейнери. Розробка мобільного застосунку детальніше описана в наступному розділі.

Висновки до розділу

Розроблені функціональна та структурна модель системи управління вивезенням відсортованого сміття.

Отримана структурна модель визначає наступні компоненти системи: контейнер зі сміттям, датчик вимірювання заповненості контейнера,

					ІТ61.070БАК.004 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дат		

центральний обробник даних, база даних з актуальними даними про заповненість всіх контейнерів, автоматизоване робоче місце адміністратора та водія. Реалізацією автоматизованого робочого місця водія сміттєвоза вибраний мобільний застосунок.

Отримана функціональна модель визначає призначення системи та описує функції системи управління вивезенням відсортованого сміття за допомогою методології IDEF0.

Визначені компоненти, які можуть використовуватися в прототипі системи управління вивезенням відсортованого сміття, а саме: сервіс Firebase, який використовується у якості центрального обробника даних, отримує дані про заповненість контейнерів зі сміттям та надає доступ до цих даних автоматизованому робочому місцю водія, Firebase Realtime Database, мікроконтролер Arduino ESP8266 IOT Elecrow та ультразвуковий датчиком відстані HC-SR04.

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		25

3 РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ СИСТЕМИ УПРАВЛІННЯ ВИВЕЗЕННЯМ ВІДСОРТОВАНОГО СМІТТЯ

У розділі проаналізована предметна область та функціональна модель. Проведений огляд платформ та інструментів для розробки мобільного застосунку. Описана реалізація програмного коду мобільного застосунку.

3.1 Призначення мобільного застосунку

Мобільний застосунок, який є частиною системи управління вивезенням відсортованого сміття, призначений для оптимізації вивезення твердих побутових відходів. Застосунок отримує актуальні дані про заповненість всіх типів смітників та будує оптимальний маршрут через заповнені смітники. Таким чином мобільний застосунок вирішує проблему вчасного вивозу твердих побутових відходів та зменшує собівартість вивозу сміття.

Цільовим користувачем проєктованого мобільного застосунку є міста, підприємства, які здійснюють вивезення твердих побутових відходів.

3.2 Аналіз предметної області мобільного застосунку

Предметною областю проєктованого мобільного застосунку є процес вивезення твердих побутових відходів, в тому числі відсортованого сміття.

Головними елементами є контейнери зі сміття, які мають такі властивості: відсоток заповненості контейнера сміттям, тип сміття, який містить контейнер (скло, пластик, змішане, тощо). Контейнери знаходяться в певних локаціях – точках вивозу, які мають координати.

Користувач застосунку здійснює збір сміття із контейнерів дотримуючись певного маршруту.

					ІТ61.070БАК.004 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дат		

3.3 Розробка функціональної моделі мобільного застосунку

Опис функцій мобільного застосунку системи управління вивезенням відсортованого сміття та взаємодія користувачів з ним зображена на діаграмі варіантів використання [10] на кресленику ІТ61.070БАК.004 Д2.

Єдиним актором мобільного застосунку системи управління вивозом відсортованого сміття є його користувач, водій сміттєвоза. Лише він взаємодіє із мобільним застосунком.

Актору доступні дві функції мобільного застосунку:

- переглянути заповненість контейнера;
- прокласти маршрут через заповнені контейнери певного типу.

Для перегляду заповненості контейнеру користувач має спочатку вибрати його на карті, на якій зображені всі смітники всіх типів, які входять до даної системи управління вивезенням відсортованого сміття.

Для прокладання маршруту через заповнені контейнери користувач має задати критерій визначення заповненості контейнерів зі сміттям, вибрати тип сміття, яке містить контейнер та вибрати область точок вивозу. В даній області будуть відбиратися точки вивозу з контейнерами, які підходять під заданий тип контейнера та рівень заповненості. Задання області відбувається за допомогою вибору початкової та кінцевої точки на карті. Через ці точки формується коло, яке визначає область точок вивозу.

3.4 Структура програмних компонентів мобільного застосунку

Структура представлення програмних сутностей мобільного застосунку системи управління вивезенням відсортованого сміття на найвищому рівні представлена на рисунку 3.1.

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		27

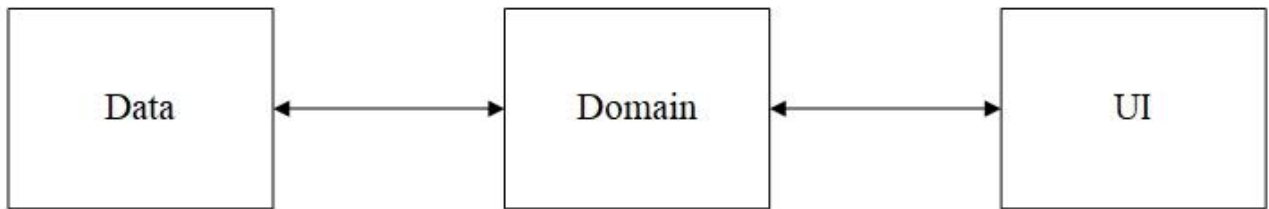


Рисунок 3.1 – Структура груп програмних компонентів мобільного застосунку

Абстрагуючись від реалізації, всі програмні сутності мобільного застосунку системи управління вивезенням відсортованого сміття можна поділити на 3 шари: UI, Domain та Data.

До шару UI відносяться всі програмні сутності, які формують графічний інтерфейс користувача, відображаючи йому інформацію мобільного застосунку. Графічний інтерфейс користувача поділяється на екрани, у якого є своє призначення.

До шару Domain відносяться всі предметні сутності мобільного застосунку та сутності, які містять бізнес-логіку мобільного застосунку. Під бізнес-логікою розуміється набір бізнес-правил мобільного застосунку, які оброблюють предметні сутності.

До шару Data відносяться програмні сутності, які відповідають за доступ даних, потрібних для функціонування мобільного застосунку. Найчастіше це доступ до даних в локальному сховищі, або сховищі, що знаходиться на сервері.

Структура екранів мобільного застосунку будується навколо шаблону MVVM (Model-View-ViewModel) [11].

Нижче представлений детальніший опис функцій компонентів шаблону MVVM.

View – компонент, обов’язком якого є відображення даних користувачу на екрані. Також цей компонент містить в собі всю логіку, яка пов’язана з відображенням. Наприклад, логіку анімацій.

ViewModel – компонент, який зберігає в собі дані, потрібні для відображення користувачу. Ці дані отримуються від Model. ViewModel приймає вхідні дані від користувача. За допомогою шаблону «Спостерігач» збережені дані надаються компоненту View.

Model – набір компонентів, які містять бізнес-логіку застосунку, та логіку доступу до даних, що зберігаються в локальному або віддаленому сховищах.

Взаємодія між компонентами зображено на рисунку 3.2.

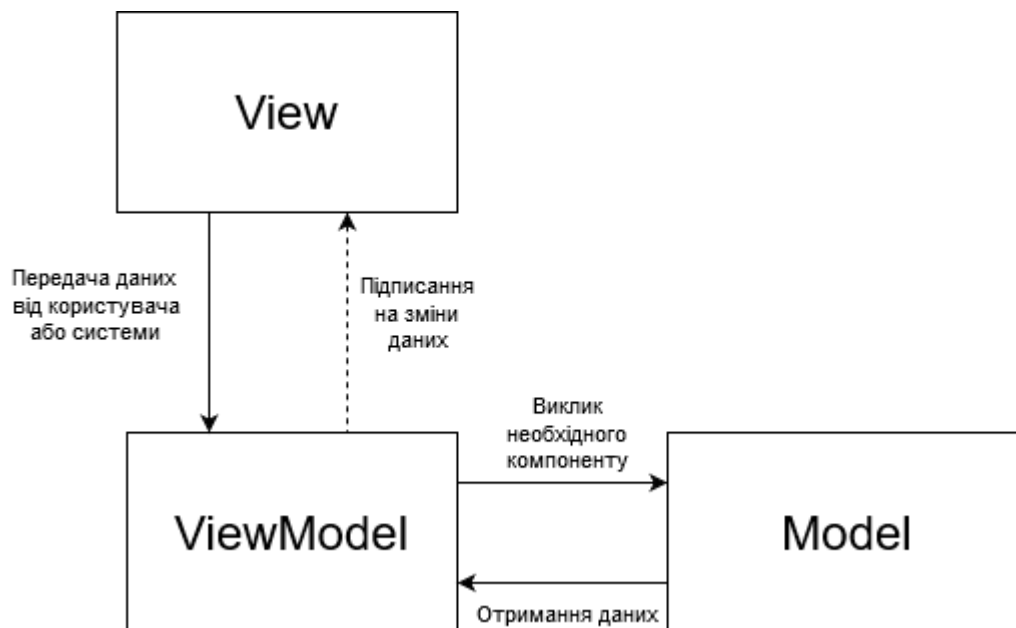


Рисунок 3.2 – Взаємодія компонентів MVVM

View надсилає компоненту ViewModel дані від користувача або системи. Далі ViewModel викликає відповідні компоненти Model та отримує дані від них. Отримані дані зберігаються всередині ViewModel, та надаються назовні через шаблон «Спостерігач» [12]. ViewModel надає можливість спостерігати за змінами конкретних даних. В свою чергу View підписується на оновлення цих даних для того, щоб відображати актуальну інформацію користувачеві на екрані. Підписка на ці дані відбувається під час ініціалізації компонента View.

Шаблон «Спостерігач» – це поведінковий шаблон проєктування, який створює механізм підписки, що дає змогу одним об’єктам стежити й реагувати на події, які відбуваються в інших об’єктах, зокрема підписуватися на зміну даних поля певного об’єкта. Шаблон складається із двох сутностей: Publisher – володіє внутрішнім станом, зміни якого можуть відслідковуватися, підписавшись на них, Subscriber – підписуються на оновлення зміни стану

Publisher, та виконують якусь дію при зміні спостережуваного стану. Uml діаграма шаблону «Спостерігач» зображена на рисунку 3.3.

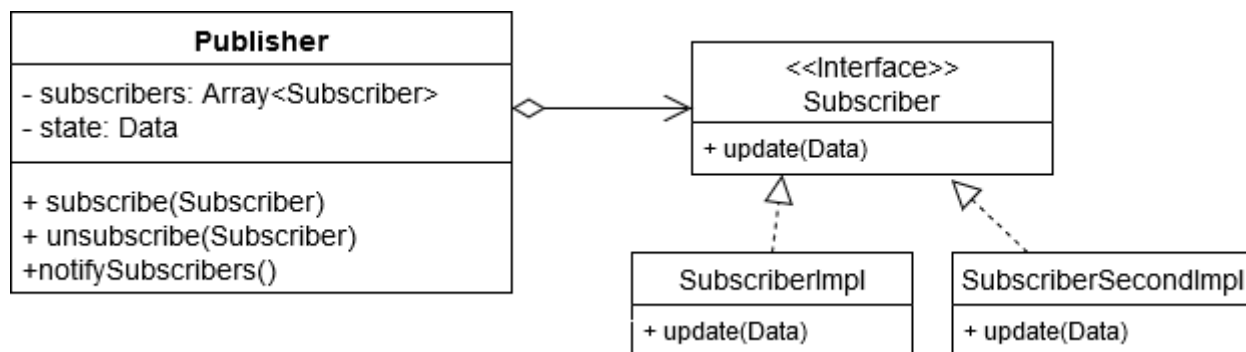


Рисунок 3.3 – Uml діаграма шаблону «Спостерігач»

Об’єкт, який хоче підписатися на зміни стану Publisher повинен реалізувати інтерфейс Subscriber. В цьому інтерфейсі міститься метод update, який викликається при зміні спостереженого стану. Для підписки на стан, об’єкт, що реалізує інтерфейс Subscriber має викликати метод subscribe у Publisher. Один Publisher може надавати підписку багатьом Subscriber.

3.5 Вибір операційної системи для мобільного застосунку

На сьогоднішній день на ринку мобільних операційних систем, які використовуються в смартфонах та планшетах є лише дві актуальні операційні системи: Android [13] та IOS [14]. Обидві операційні системи зараз активно розвиваються.

3.5.1 Операційна система Android

Операційна система Android на сьогоднішній день є найпоширенішою системою серед сучасних смартфонів та планшетів. Доля Android на ринку операційних систем перевищує 80% у 2019 році. Лише деякі виробники смартфонів та планшетів не використовують її в своїх пристроях.

Операційна система розроблюється компанією Google. Вихідний код ядра операційної системи знаходиться у відкритому доступі, що є частиною AOSP (Android open source project). Це дає можливість виробникам пристроїв адаптувати операційну систему під свої пристрої та покращувати операційну систему. На сьогоднішній день остання випущена версія операційної системи – Android 10.

Популярність операційної системи можна пояснити тим, що на її основі виходять велика кількість пристроїв, причому не лише смартфонів та планшетів, а й смарт-годинники, велосипеди, телевізори й навіть деякі автомобілі. В свою чергу операційна система Android одночасно унікальна та різноманітна – кожен користувач може налаштовувати графічну оболонку операційної системи. Сама операційна система є безкоштовною.

Переваги операційної системи Android:

- відкритий вихідний код ядра операційної системи (AOSP);
- зручність передачі даних між пристроями;
- зручне та інтуїтивне управління;
- широкий вибір пристроїв від різних виробників;
- доступність інструментів розробки на будь-яких платформах;
- безкоштовна;
- підтримує розробку на мовах програмування Java [15] або Kotlin [16];
- має широку спільноту розробників.

До недоліків можна віднести:

- широкий набір пристроїв з різними конфігураціями, тобто різною діагоналлю екранів, співвідношенням сторін;
- наявність виробників, які не оптимізують операційну систему під свої пристрої;
- Велика кількість старих версій операційної системи, які потрібно підтримувати.

3.5.2 Операційна система IOS

Операційна система IOS розроблюється компанією Apple для своїх мобільних пристроїв: смартфонів Iphone та планшетів Ipad. Компанія одночасно є як виробником смартфонів та планшетів, так і розробником операційної системи для неї. Такий підхід може гарантувати найвищу якість та надійність своїх пристроїв, що в свою чергу позначається на ціні. На сьогодні актуальною версією операційної системи є версія IOS 13.

Переваги операційної системи IOS:

- простий та продуманий інтерфейс;
- зручне та інтуїтивне управління;
- значно менше підтримуваних старих версій операційної системи;
- широкий вибір мобільних застосунків;
- жорстка перевірка мобільних застосунків в App Store;
- підтримує розробку на мовах програмування Swift [17] або

Objective-C [18];

- має широку спільноту розробників.

До недоліків можна віднести:

- висока вартість пристроїв разом з операційною системою;
- обмежений вибір пристроїв для користувачів;
- інструменти розробки мобільних застосунків лише на macOS;
- повністю закрита операційна система.

3.5.3 Результат аналізу операційних систем

Результат порівняння ключових можливостей операційних систем Android та IOS для розробки мобільних застосунків зображено в таблиці 3.1.

					IT61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		32

Таблиця 3.1 – Порівняння можливостей операційних систем

Назва	Android	IOS
Доля на ринку мобільних пристроїв на 2019 рік	Приблизно 80%	Приблизно 19%
Мови програмування	Java або Kotlin	Objective-C або Swift
Доступність інструментів розробки	На будь-яких платформах	macOS
Пристрої, на яких працює операційна система	Смартфони, планшети, телевізори, годинники	Смартфони, планшети
Вартість пристроїв, які використовують операційну систему на 23.05.2020	Від 1000 грн.	Від 10000 грн

Операційна система Android займає більшу частину ринку мобільних пристроїв через широку різноманітність пристроїв в різних цінових діапазонах, починаючи від найдешевших пристроїв. В свою чергу найдешевший пристрій на операційній системі IOS коштують значно дорожче ніж на операційній системі Android. Обидві операційні системи мають зручні інструменти та підтримують сучасні мови програмування для розробки мобільних застосунків.

Розробка мобільного застосунку системи управління вивезенням відсортованого сміття в рамках цього дипломного проєкту проводилась на операційній системі Android. Вибір зумовлений великою кількістю пристроїв, що працюють на цій операційній системі та велика доступність інструментів розробки мобільних застосунків під цю операційну систему. В майбутньому, мобільний застосунок системи управління вивезенням відсортованого сміття може бути написаний і під операційну систему IOS.

3.6 Вибір мов програмування для розробки мобільного застосунку

Розробку мобільного застосунку під операційну систему Android можна здійснювати за допомогою двох мов програмування: Java та Kotlin. Обидві мови програмування активно підтримуються та розвиваються.

3.6.1 Мова програмування Java

Java – статично типізованою, об’єктно орієнтованою мовою програмування. Була розроблена в 1995 році компанією Sun Microsystems. Зараз мова програмування підтримується компанією Oracle. Програмний код, написаний на цій мові програмування, транслюється в проміжний код, так званий байт-код, який далі виконується віртуальною машиною, спеціально написаною під конкретну платформу. В операційній системі Android віртуальна машина називається Dalvik.

До складу Java входить Java Development Kit – набір розробника зі стандартними бібліотеками класів Java, документація цих класів та приклади.

Окрім використання Java як мови програмування для розробки мобільних застосунків під операційну систему Android, її також використовують для розробки веб-застосунків, технологій обробки великих даних, серверних застосунків.

Переваги мови програмування Java є:

- безпечний – виконання написаної програми всередині віртуальної машини;
- простий синтаксис;
- великий набір стандартних класів для розробки (колекцій, алгоритмів);
- має широку спільноту розробників;
- можливість писати платформно незалежний код.

Недоліки мови програмування Java:

- необхідність писати багато шаблонного коду, наприклад, для

					IT61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		34

створення Plain Old Java Objects;

- незручно працювати з багатопотоковим кодом;
- проблема з null-об'єктами;
- не всі старі версії операційної системи Android підтримують нові функції Java.

3.6.2 Мова програмування Kotlin

Kotlin – статично типізована мова програмування загального призначення, розроблюється компанією JetBrains. Сама мова програмування не прив'язана до якоїсь конкретної парадигми програмування. Ця мова програмування була розроблена для взаємодії з Java та JVM. Реліз версії Kotlin 1.0 відбувся на початку 2016 року. У 2017 компанія Google оголосила підтримку цієї мови для розробки мобільних застосунків під операційну платформу Android. З 2019 Google рекомендує надавати перевагу мові Kotlin при розробці нових мобільних застосунків на операційній системі Android.

Однією з головних особливостей мови є визначення типу змінної по її значенню, що робить написаний програмний код значно лаконічним. Ще однією особливістю мови програмування є nullable типи даних. Всі типи даних в Kotlin діляться на два типи: ті що можуть містити посилання на null та ті що не можуть. Таким чином компілятор може відслідковувати всі місця, де може виникнути ситуація з отриманням NullPointerException.

Написані програми на Kotlin можуть працювати не лише на платформах з JVM. Проєкт Kotlin/JS дозволяє транслювати написаний код на мові програмування Kotlin в JS для створення веб-застосунків. Проєкт Kotlin Multiplatform, який дозволяє компілювати код, написаний на цій мові в машинний код різних платформ, зокрема IOS, Windows, Linux, Assembly.

Для роботи з асинхронним виконанням програмного коду в Kotlin є фреймворк Kotlin Coroutines. Цей фреймворк базується на концепції відкладених обчислень. Суть концепції полягає в тому, щоб не блокувати потоки виконання

					IT61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		35

програми при виконанні якоїсь довготривалої роботи, наприклад, очікування відповіді на запит від сервера. Іншими словами Kotlin Coroutines – це легкі потоки, які виконують якусь роботу асинхронно, не блокуючи потоків процесу. Такий підхід значно пришвидшує роботу застосунків та полегшує написання програмного коду розробникам.

Переваги мови програмування Kotlin є:

- повна сумісність з мовою Java та її платформою. Всі бібліотеки та фреймворки, що працюють на Java, працюють на Kotlin;
- наявність зручного інструменту для роботи з асинхронним програмним кодом (Kotlin Coroutines);
- nullable типи;
- рекомендується компанією Google для використання у якості мови програмування для розробки мобільних застосунків для операційної системи Android.

3.6.3 Результат огляду мов програмування для розробки мобільного застосунку

Провівши огляд мов програмування для розробки мобільного застосунку для операційної системи Android, вибрана мова програмування Kotlin для розробки мобільного застосунку системи управління вивезення відсортованого сміття на операційній системі Android в рамках бакалаврського проєкту. Kotlin має всі можливості що і мова програмування Java. Окрім цього мова програмування Kotlin є більш лаконічною ніж мова програмування Java. Наявність Kotlin Coroutines дозволяє простіше працювати з асинхронним програмним кодом, в той час як в мові програмування Java немає таких аналогів. Kotlin є рекомендованою мовою програмування для розробки мобільних застосунків під операційну систему Android компанією Google.

					IT61.070БАК.004 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дат		

3.7 Вибір картографічних сервісів та сервісів побудови маршрутів для розробки мобільного застосунку

Головна функція мобільного застосунку системи управління вивезенням відсортованого сміття є побудова маршрутів через точки вивозу сміття. Для побудови таких маршрутів використовуються спеціальні картографічні сервіси. Окрім побудови маршрутів застосунку потрібно отримувати деяку інформацію про точки вивозу сміття, наприклад назву вулиці, де розташована точка вивозу. Отримання даних про певне місце на карті використовуючи географічні координати цього місця називається геокодування. Здійснюють його картографічні сервіси.

3.7.1 Сервіс Google Maps

Google Maps [19] – це картографічний сервіс та набір технологій пов'язаних з цим сервісом, яка розроблена компанією Google. Картографічний сервіс легко інтегрується з мобільними застосунками на операційній системі Android надаючи Maps SDK для цієї операційної системи. Maps SDK самостійно оброблює та надсилає запити до Google Maps Api, надаючи розробнику зручний програмний інтерфейс. Окрім цього Google Maps надає графічний елемент відображення карти, який може бути доданий до графічного інтерфейсу користувача для відображення карти. На цій карті можна додавати маркери, полігони, змінювати вид карти та просто взаємодіяти з нею.

Окрім картографії сервіс надає інструмент для побудови маршрутів між заданими точками через Directions API. Сервіс будує оптимальний маршрут по заданому напрямленню. Побудова маршруту може здійснюватися для різних типів транспорту.

Недоліком сервісу Google Maps є відносно дорога вартість сервісу. Використання послуг геокодування коштує п'ять доларів за 1000 запитів.

					ІТ61.070БАК.004 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дат		

Використання послуги побудови маршруту коштує також п'ять доларів за 1000 запитів.

3.7.2 Сервіси Here

Сервіс Here [20] – набір картографічних сервісів та сервісів, що працюють з ними. Розроблені компанією Here Technologies. До складу набору входять такі сервіси:

- Maps – картографічний сервіс, являє собою набір карт;
- Geocoding – сервіс геокодування, тобто отримання інформації про місце за графічними координатами цього місця і навпаки;
- Search – пошук місця за його назвою на карті;
- Routing – прокладання маршрутів через задані точки.

Сервіс Routing надає можливість прокладати оптимальні маршрути через задані точки з урахування заторів та об'їздів. Окрім того Routing має функцію прокладання маршруту для різних типів транспорту.

Перевагами Here є безкоштовний доступ до всіх можливостей всіх сервісів, але з певними обмеженнями.

До недоліків можна віднести місцями неточне визначення назв місць або відсутність цих місць в Україні.

Here надає набір програмних компонентів Here SDK Lite для операційної системи Android. Компоненти написані на мові програмування Java. За допомогою Here SDK Lite розробник мобільного застосунку на операційній системі Android може взаємодіяти зі всіма сервісами Here.

3.7.3 Сервіс OpenStreetMaps

Сервіс OpenStreetMaps [21] – некомерційний картографічний сервіс, який є проєктом з відкритим кодом. Розробка проводиться ентузіастами. Дані до цього сервісу вносять місцеві користувачі.

					ІТ61.070БАК.004 ПЗ	Арк.
						38
Змн.	Арк.	№ докum.	Підпис	Дат		

Сервіс містить в собі дані про назви доріг, вулиць, міст і містечок та багато іншого. Користуватися даними цього сервісу можна безкоштовно.

OpenStretMaps містить функцію прокладання оптимального маршруту через дві точки.

Недоліком даного сервісу є відсутність підтримуваної SDK для операційної системи Android та відсутність сервісу побудови маршрутів через декілька точок.

3.7.4 Результат аналізу картографічних сервісів та сервісів побудови маршрутів для розробки мобільного застосунку

Результат порівняння ключових можливостей картографічних сервісів та сервісів побудови маршрутів для розробки мобільного застосунку на операційній системі Android зображено в таблиці 3.2.

Таблиця 3.2 – Порівняння можливостей картографічних сервісів та сервісів побудови маршрутів

Назва	Google Maps	Here	Open Street Maps
Наявність картографічного сервісу	Присутній	Присутній	Присутній
Наявність сервісу побудови маршрутів	Присутній	Присутній	Відсутній
Наявність безкоштовної версії	Відсутній	Присутній, але з деякими обмеженнями	Присутній
Ціна	Геокодування – 5\$ за 1000 запитів,	Безкоштовно 250 000 запитів в місяць або до	Безкоштовно

Назва	Google Maps	Here	Open Street Maps
	побудови маршруту – 5\$ за 1000	5000 активних користувачів	
Наявність SDK для Android	Присутній	Присутній	Відсутній

Отже, провівши огляд наявних можливостей картографічних сервісів та сервісів побудови маршрутів для розробки мобільного застосунку на операційній системі Android, був вибраний набір сервісів Here для розробки в рамках цього дипломного проєкту. Here має наявність SDK для Android, що дуже сильно спрощує інтеграцію сервісів Here з мобільним застосунком. Мобільний застосунок використовуватиме сервіс Here Maps для відображення карти користувачу та сервіс Here Routing для побудови маршрутів через задані точки. Окрім цього головним фактором вибору цього набору сервісів є його безкоштовний доступ на 250 000 запитів в місяць або до 5000 активних користувачів мобільного застосунку, що є цілком достатньо для розроблюваного мобільного застосунку.

3.8 Середовище розробки мобільного застосунку

Розробка мобільного застосунку на операційній системі Android проводиться в інтегрованому середовищі розробки Android Studio [22]. Це середовище розробки підтримує мови програмування Kotlin та Java. Саме середовище було розроблюється компанією Google з метою надання зручного інструменту для розробки мобільних застосунків під операційну систему Android. Android Studio базується на основі інтегрованого середовища розробки IntelliJ IDEA від компанії JetBrains. Окрім доступності до всіх функцій середовища IntelliJ IDEA, Android Studio має шаблони для створення програмних компонентів мобільних застосунків операційної системи Android, графічний

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		40

редактор розробки інтерфейсу користувача мобільного застосунку, інтеграцію із системою ProGuard та Android емулятор всіх версій.

Взаємодія з функціями операційної системи проводиться за допомогою Android SDK, що містить набір класів та програмних інтерфейсів.

3.9 Використані методи та алгоритми при розробці мобільного застосунку

Під час написання розробки мобільного застосунку системи управління вивезенням відсортованого сміття був використаний метод виявлення точок, що входять в круг між двома заданими точками та алгоритм сортування точок за наближеністю до заданої точки.

Метод фільтрування точок фільтрує всі точок, пропускаючи лише ті, що входять в круг, побудований між двома заданими точками користувачем.

Алгоритм сортування точок за наближеністю до заданої точки сортує всі точки відносно початкової та кінцевої точок заданих користувачем. Результатом роботи алгоритму є масив відсортованих точок, які розташовані від найближчих до початкової точки, до найближчих до кінцевої точки.

3.9.1 Метод виявлення точок, що входять в круг між двома точками

Вхідними даними задачі є початкова та кінцева точки задані користувачем. Нехай початкова точка – $A(x_a, y_a)$, кінцева точка – $B(x_b, y_b)$. Тоді знайдемо середню точку $M(x_m, y_m)$, яка буде центром круга за формулою:

$$M\left(\frac{x_a+y_a}{2}, \frac{x_b+y_b}{2}\right). \quad (3.1)$$

Далі розрахуємо радіус круга R за формулою:

$$R = \sqrt{(x_m - x_a)^2 + (y_m - y_a)^2}. \quad (3.2)$$

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		41

Маючи центр та радіус можемо записати нерівність, яка буде математичним представленням круга:

$$(x - x_m)^2 + (y - y_m)^2 < R^2. \quad (3.3)$$

Використовуючи нерівність 3.3 ми можемо виявляти точки, які потрапляють в цей круг підставляючи їх координати замість x та y .

3.9.2 Алгоритм сортування точок за наближеністю до заданої

Для задачі сортування точок від тих, що найближче знаходяться до початкової точки до найвіддаленіших підходить алгоритм, розроблений за підходом «Розділяй та володарюй» [23], тобто рекурсивному розбитті розв’язуваної задачі і злиття їх рішення для вирішення завдання. На основі прикладу такого підходу – алгоритму сортування злиттям, був розроблений адаптований алгоритм сортування точок від тих, що найближче знаходяться до початкової точки до найвіддаленіших.

Суть роботи алгоритму полягає в наступному: знаходимо середню точку між початковою та кінцевою точками, які являються крайніми точками. Розділяємо масив точок на два підмасиви: перший – точки, які знаходяться ближче до початкової точки, другий – точки, які знаходяться ближче до кінцевої точки. Тепер крайніми точками першого підмасиву будуть початкова точка та середня точка. Крайніми точками другого масиву будуть середня точка та кінцева точка. Далі аналогічно робимо для кожного підмасиву, аж поки в підмасиві не залишиться одна або менше точок. Далі підмасиви зливаються й ми отримуємо відсортований масив точок від тих, що найближче знаходяться до початкової точки до найвіддаленіших.

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		42

3.10 Структура вихідного коду мобільного застосунку

Оскільки розробка мобільного застосунку системи управління вивезенням відсортованого сміття здійснюється за допомогою мови програмування Kotlin, програмний код організовано у вигляді об'єктів відповідних класів, які взаємодіють між собою. Ці класи розташовані у відповідних файлах, а файли розміщені по пакетах.

Вихідний код мобільного застосунку системи управління вивезенням відсортованого сміття розташований в кореневій директорії та складається з наступних пакетів:

- controller;
- data;
- di;
- navigation;
- services;
- ui.

Окрім пакетів там розташовані файли з класами:

- App – клас, який містить в собі програмний код для налаштування мобільного застосунку перед запуском та ініціалізації його компонентів;
- MainActivity – представляє клас, який є головним екраном мобільного застосунку, створює та відображає графічний інтерфейс користувачу, що описаний у відповідному xml файлі.

В пакеті controller знаходиться клас:

- LocationPermissionCallbackController – перевіряє наявність прав на отримання місцезнаходження девайсу при відкритті кожного екрану застосунку.

В пакеті data знаходяться наступні пакети:

- model – містить класи, що представляють дані в цьому застосунку;
- repository – містить логіку доступу до цих даних;
- event – містить інтерфейс та базову реалізацію опису подій в мобільному застосунку.

					ІТ61.070БАК.004 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дат		

Пакет model містить такі класи:

- Bin – представляє дані заповненості смітника;
- BinCoordinate – представляє координати місцеположення смітника;
- BinType – представляє тип смітника;
- Circle – представляє математичний опис круга;
- MapMarkModel – представляє дані маркера на карті.

Пакет repository містить такі інтерфейси:

- BinsRepository – описує інтерфейс доступу до даних смітників. Реалізує цей інтерфейс клас BinRepositoryImpl;
- MapRepository – описує інтерфейс доступу до даних карти, таких як адреса вулиць. Реалізує інтерфейс клас MapRepositoryImpl.

Окрім інтерфейсів, в пакеті repository лежить пакет impl, який містить класи реалізації відповідних інтерфейсів.

Пакет navigation містить класи та інтерфейси:

- DerictionNavigationRoute – базова реалізація інтерфейсу NavigationRoute;
- NavigationRoute – загальний інтерфейс для навігації всередині застосунку.

Пакет services містить загальний інтерфейси, та реалізацію бізнес-логіки застосунку. Всередині пакету знаходяться наступні пакети:

- binCoordinates;
- location;
- mapRouting;

Пакет binCoordinates містить інтерфейс BinCoordinatesService, що описує методи підготовки координат смітників до побудови маршруту через них та клас реалізацію цього інтерфейсу BinCoordinatesServiceImpl.

Пакет location містить інтерфейс UserLocationService, що описує методи отримання поточних координат пристрою та клас реалізацію цього інтерфейсу UserLocationServiceImpl.

					ІТ61.070БАК.004 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дат		

Пакет `mapRouting` містить інтерфейс `MapRoutingService`, що описує методи прокладання маршруту через задані точки та клас реалізацію цього інтерфейсу `MapRoutingServiceImpl`.

Пакет `ui` містить наступні пакети:

- `selectDestinations`;
- `map`;
- `route`;
- `bin`;
- `base`;
- `ext`.

Пакет `base` містить базові класи:

- `BaseFragment` – базовий клас для частин екрану графічного інтерфейсу користувача;
- `EventableViewModel` - базовий `ViewModel` клас, який може підписуватися на події мобільного застосунку.

Пакет `ext` містить файл з функціями, які містять допоміжний код для роботи з елементами відображення.

Пакет `map` містить наступні пакети:

- `bin`;
- `navigation`;
- `selectDestinations`;
- `view`;
- `viewModel`.

Пакет `view`, що міститься в пакеті `map`, включає класи: `MapFragment` – клас, який містить програмний код для відображення графічної карти користувачу, `MapMarkersAdapter` – клас, який адаптує дані про розміщення маркерів на карті для їх відображення на екрані.

Пакет `viewModel`, що міститься в пакеті `map`, включає клас `MapViewModel`, який зберігає дані, що потрібно відобразити користувачу на відповідному екрані.

					IT61.070БАК.004 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дат		

Пакет navigation, що міститься в пакеті map, містить клас SelectDestinationRoute, перехід на екран вибору точок вивозу сміття.

Пакет bin містить наступні пакети:

- view;
- viewModel.

Пакет view, що міститься в пакеті bin, включає клас BinDetailBottomSheetDialogFragment – клас, який містить програмний код для відображення даних про заповненість контейнера користувачу.

Пакет viewModel, що міститься в пакеті bin, включає клас BinDetailViewModel, який зберігає дані, що потрібно відобразити користувачу на відповідному екрані.

Пакет selectDestinations містить наступні пакети:

- navigation;
- sharedViewModel;
- view;
- viewModel.

Пакет view, що міститься в пакеті selectDestinations, включає клас SelectDestinationsFragment – клас, який містить програмний код для вибору точок вивозу сміття користувачу.

Пакет viewModel, що міститься в пакеті selectDestinations, включає клас SelectDestinationsViewModel, який зберігає дані, що потрібно відобразити користувачу на відповідному екрані.

Пакет navigation, що міститься в пакеті selectDestinations, містить клас MapRoute, перехід на екран відображення карти.

Пакет sharedViewModel, що міститься в пакеті selectDestinations, включає клас MapDestinationSharedViewModel – клас, який зберігає дані про вибрані точки користувачем для відображення на різних графічних елементах користувачу.

3.11 Структура графічних ресурсів мобільного застосунку

Графічний інтерфейс користувача в мобільному застосунку системи управління вивезенням відсортованого сміття описується за допомогою xml-файлів. Елементи графічного інтерфейсу такі як: кольори, анімації, іконки і тому подібне описуються за допомогою xml-файлів теж.

Мобільний застосунок містить всі ресурс в директорії res. До цієї директорії входить наступні директорії:

- drawable – містить іконки мобільного застосунку;
- drawable-v24 – містить іконки мобільного застосунку для версії Android 7 і вище;
- layout – містить розмітку для всіх екранів мобільного застосунку;
- mipmap – містить головну іконку мобільного застосунку;
- navigation – містить шляхи переходу між екранами;
- values – містить значення кольорів, стилів, та рядків.

3.12 Діаграми класів екранів мобільного застосунку

Програмний код мобільного застосунку системи управління вивезенням відсортованого сміття розбитий на об'єкти, які взаємодіють між собою. Кожна можливість, яку мобільний застосунок надає користувачу, належить певному екрану з графічним інтерфейсом користувача та формується за допомогою певного набору таких компонентів, що взаємодіють між собою. Це зроблено для того, щоб зробити програмний код гнучким до змін.

На рисунку 3.4 зображена діаграма класів, взаємодія яких дає можливість користувачу переглядати карту, та точки вивозу на ній.

					ІТ61.070БАК.004 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дат		

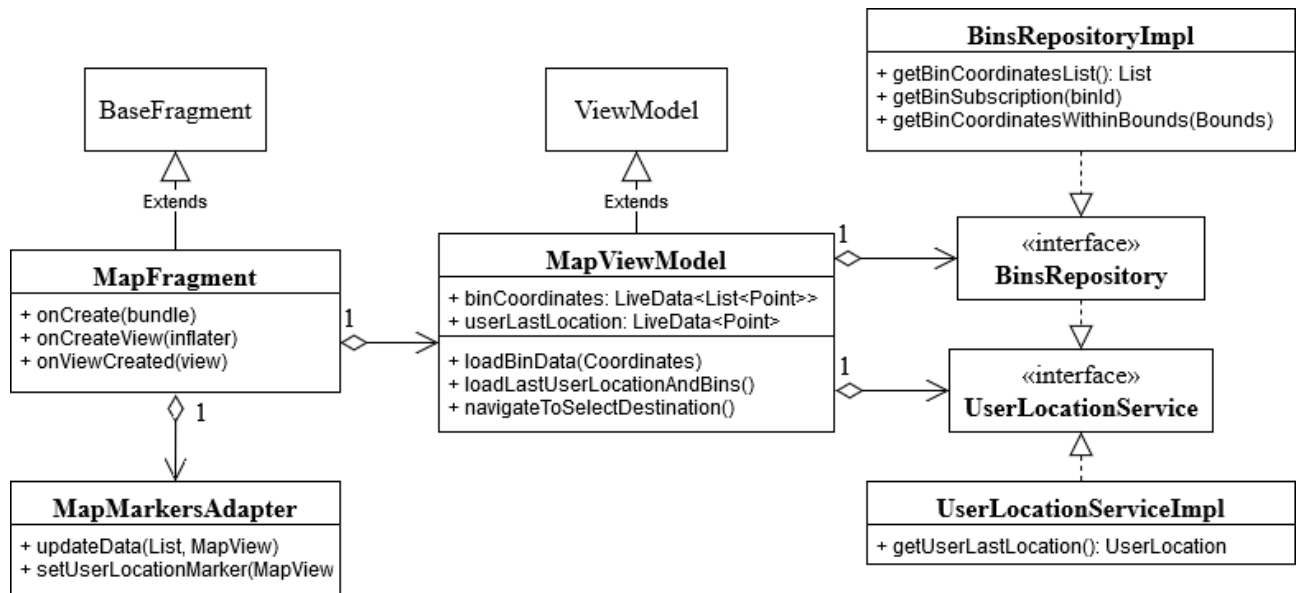


Рисунок 3.4 – UML діаграма класів перегляду карти користувачем

Користувач взаємодіє з екраном, на якому зображена карта. Програмний код цього екрану міститься в об'єкті класу MapFragment. Клас наслідує BaseFragment. Клас MapFragment містить об'єкти класу MapMarkersAdapter та класу MapViewModel.

Клас MapMarkersAdapter містить логіку адаптування даних про позиції маркерів до їх відображення на карті. Сам візуальний об'єкт карти також є частиною MapFragment.

Клас MapViewModel зберігає список точок вивозу та останнє місцезнаходження користувача. Ці дані надаються через об'єкт LiveData, який дозволяє підписуватися на зміну цих даних. Сам об'єкт MapViewModel розширює базовий клас ViewModel. Об'єкт MapViewModel містить об'єкти, які реалізують інтерфейс BinsRepository та UserLocationService.

Об'єкти, що реалізують інтерфейс BinsRepository надають доступ до списку даних контейнерів та їх точок вивозу. Реалізує цей інтерфейс клас BinsRepositoryImpl.

Об'єкти, що реалізують інтерфейс UserLocationService дають доступ до останнього місцезнаходження користувача. Реалізує цей інтерфейс клас UserLocationServiceImpl.

На рисунку 3.5 зображена діаграма класів, взаємодія яких дає можливість користувачу вибрати початкову та кінцеву точки та почати будувати маршрут через ці точки.

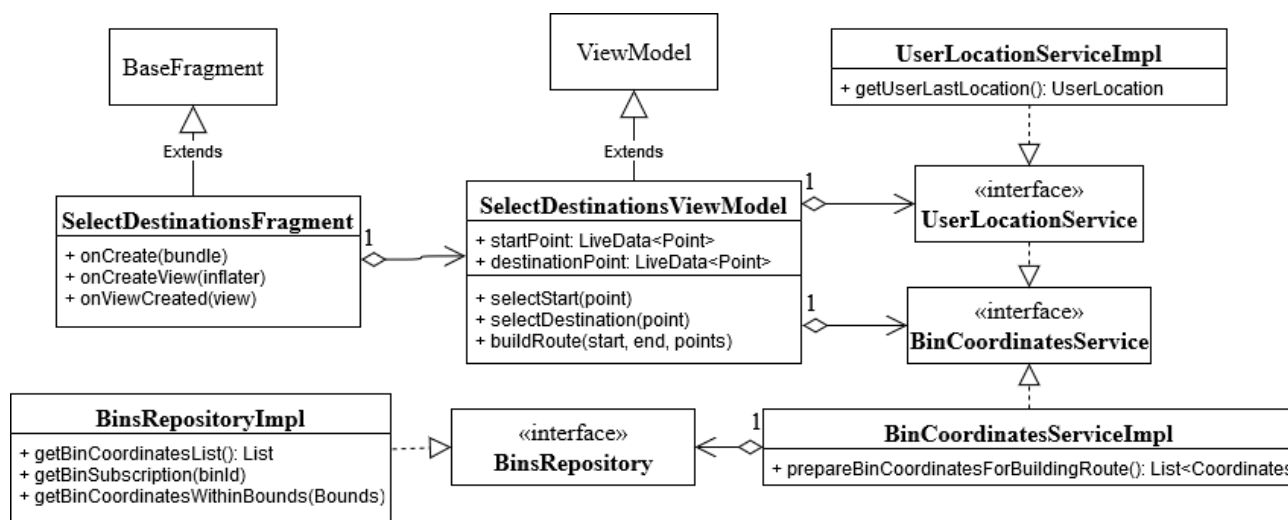


Рисунок 3.5 – UML діаграма класів можливості вибрати початкову та кінцеву точки та почати будувати маршрут через ці точки

Можливість представляє собою екран, на якому зображені два поля: початкова точка та кінцева точка, та кнопка побудувати маршрут. Програмний код цього екрану міститься в об'єкті класу SelectDestinationsFragment. Клас наслідує BaseFragment. Клас SelectDestinationsFragment містить об'єкт класу SelectDestinationsViewModel.

Клас SelectDestinationsViewModel зберігає початкову точку та останню точку, що вибрані користувачем. Ці дані надаються через об'єкт LiveData, який дозволяє підписуватися на зміну цих даних. Сам об'єкт MapViewModel розширює базовий класу ViewModel. Об'єкт MapViewModel містить об'єкти, які реалізують інтерфейс BinCoordinatesService та UserLocationService.

Об'єкти, що реалізують інтерфейс BinCoordinatesService містять логіку фільтрування та сортування точок, готуючи ці точки до прокладання маршруту через них. Реалізує цей інтерфейс клас BinCoordinatesServiceImpl, який містить об'єкт, що реалізує інтерфейс BinsRepository.

На рисунку 3.6 зображена діаграма класів, взаємодія яких дає можливість переглядати побудований маршрут на карті, та положення користувача на цьому маршруті.

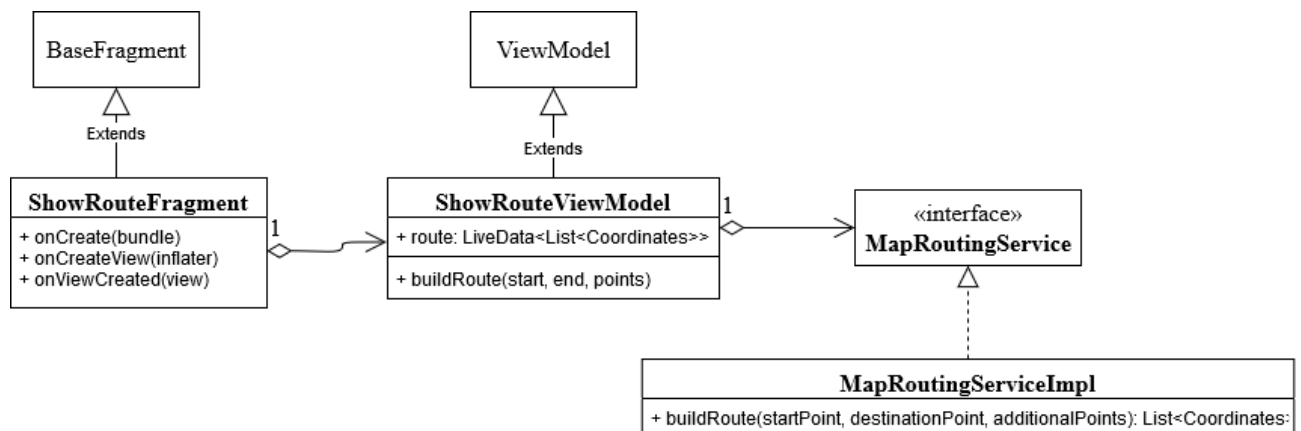


Рисунок 3.6 – UML діаграма класів можливості переглядати побудований маршрут на карті, та положення користувача на цьому маршруті

На екрані, програмний код якого міститься в класі **ShowRouteFragment**, зображена карта з прокладеним маршрутом. Об'єкт **ShowRouteFragment** містить в собі об'єкт класу **ShowRouteViewModel**, який зберігає прокладений маршрут. Цей маршрут надається через об'єкт **LiveData**, який дозволяє підписуватися на зміну маршруту. Об'єкт **ShowRouteViewModel** містить в собі об'єкт, який реалізує інтерфейс **MapRoutingService**. Об'єкти, що реалізують цей інтерфейс здійснюють побудову маршруту. Реалізує цей інтерфейс клас **MapRoutingServiceImpl**.

На рисунку 3.7 зображена діаграма класів, взаємодія яких дає можливість переглядати актуальну інформацію про контейнер в точці вивозу.

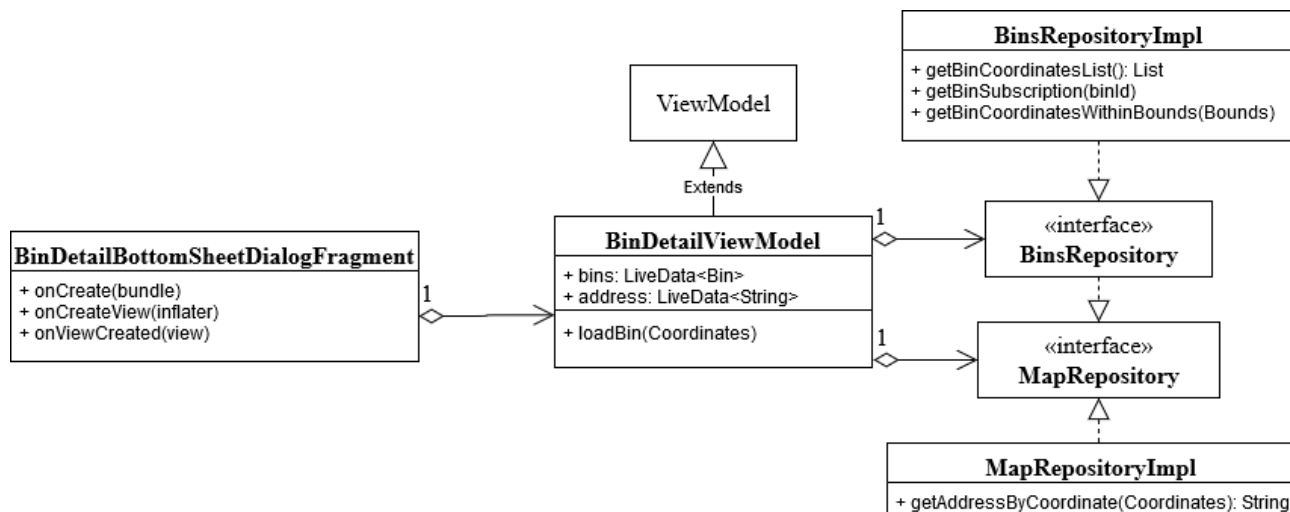


Рисунок 3.7 – UML діаграма класів можливості переглядати актуальну інформацію про контейнер

Об'єкт BinDetailBottomSheetDialogFragment містить код екрана, який відображає актуальну інформацію про контейнер та містить в собі об'єкт класу BinDetailViewModel, який зберігає дані про контейнер, які надаються через об'єкт LiveData, що дозволяє підписуватися на зміну маршруту. Об'єкт BinDetailViewModel містить в собі об'єкти, які реалізують інтерфейс BinsRepository та MapsRepository.

3.13 Діаграма послідовності роботи компонентів системи управління вивезенням відсортованого сміття

Діаграма послідовності стандартної роботи компонентів системи управління вивезенням відсортованого сміття зображена на кресленику IT61.070БАК.004 ДЗ.

Користувач системи взаємодіє лише з мобільним застосунком. Не залежно від користувача датчики відправляють інформацію про заповненість контейнерів на сервер Firebase.

Запускаючи застосунок, користувач переходить на екран з картою. Мобільний застосунок робить запит на Firebase для отримання списку точок

вивозу контейнерів зі сміттям. Після цього карта з точками вивозу відображається користувачу.

Далі користувач вибирає точки, через які мобільний застосунок повинен буде будувати маршрут. Після цього користувач натискає кнопку побудувати маршрут. Мобільний застосунок робить запит на Firebase для отримання списку заповненості контейнерів зі сміттям. Далі мобільний застосунок готує точки до побудови маршруту через них та відправляє їх сервісу Here Routing. Сервіс у відповідь відправляє побудований маршрут, який мобільний застосунок відображає користувачу.

Висновки до розділу

Розроблений мобільний застосунок системи управління вивезенням відсортованого сміття на операційній системі Android. Мобільний застосунок дозволяє переглядати актуальні дані про заповненість контейнерів зі сміттям у точках вивозу та будувати маршрути лише через точки вивозу із заповненими контейнерами. Розробка мобільного застосунку проведена з використанням мови програмування Kotlin, Firebase SDK, Android SDK та сервісу Here.

					IT61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		52

4 РОЗРОБКА КЕРІВНИЦТВА КОРИСТУВАЧА МОБІЛЬНОГО ЗАСТОСУНКУ

У розділі описані системні вимоги до пристрою, на якому має працювати мобільний застосунок системи управління вивезенням відсортованого сміття та розроблене керівництво користувача.

4.1 Визначення системних вимог пристрою для запуску мобільного застосунку

Мобільний застосунок системи управління вивезенням відсортованого сміття має такі системні вимоги:

- пристрій на базі операційної системи Android версії 5 або вище;
- підключення до мережі Інтернет;
- наявність сервісів Google Play [24];
- модуль GPS.

Google play сервіси – системний застосунок для операційної системи Android від Google, який виконує багато функцій. Однією з них є надання якісного доступу мобільним застосункам до даних про поточне місцезнаходження пристрою.

4.2 Керівництво користувача

Відкриваючи мобільний застосунок вперше, користувач бачить діалог, на якому мобільний застосунок просить надати йому доступ до отримання поточного місцезнаходження пристрою. Операційна система Android не дає мобільним застосункам доступ до чутливої інформації, зокрема інформації про місцезнаходження пристрою, без дозволу користувача.

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		53

На діалозі зображений текст, який означає, що мобільний застосунок системи управління вивезенням відсортованого сміття просить доступ до місцезнаходження пристрою та дві кнопки: відхилити та прийняти.

У випадку, якщо користувач не надає доступ до місцезнаходження, застосунок показує йому екран з поясненням, що даний мобільний застосунок не може працювати без доступу до поточного місцезнаходження пристрою та при кожному наступному відкриванні застосунку знову буде показувати діалог.

Діалог зображений на рисунку 4.1.

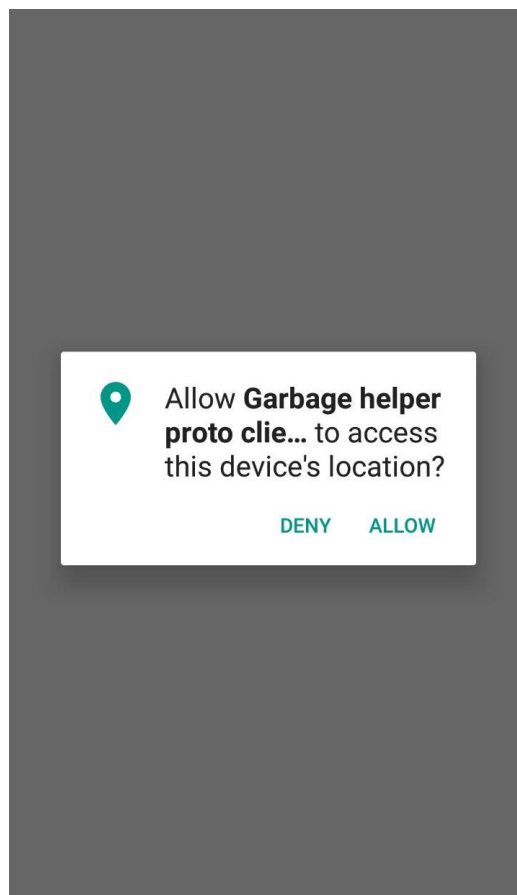


Рисунок 4.1 – Діалог мобільного застосунку, щодо надання доступу до місцезнаходження пристрою

Надаючи доступ до поточного місцезнаходження пристрою користувач потрапляє на головний екран мобільного застосунку. На екрані зображена карта, на якій відображається поточне місцезнаходження користувача та точки вивозу

контейнерів зі сміттям. В правому нижньому куті екрану знаходиться кнопка вибору точок, для побудови маршруту.

В подальшому, при відкритті мобільного застосунку буде відкриватися саме цей екран.

Головний екран мобільного застосунку системи управління вивезенням відсортованого сміття зображений на рисунку 4.2.

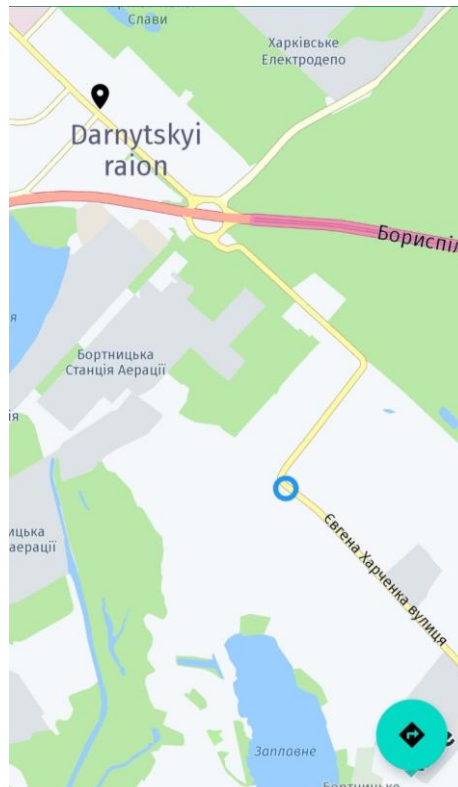
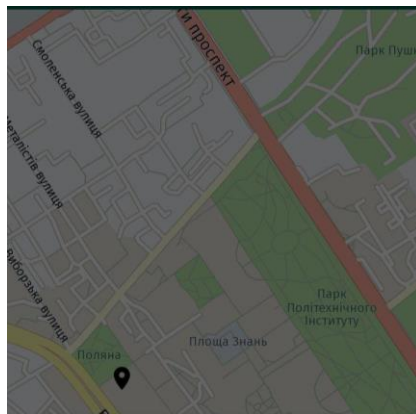


Рисунок 4.2 – Головний екран мобільного застосунку

Натискаючи на точку вивозу контейнера зі сміттям користувачу відображається інформація про цей контейнер: рівень заповненості, тип контейнера, вулиця, на якій розташований контейнер. При зміні заповненості контейнера застосунок одразу отримує актуальні дані про заповненість та образу відображає їх користувачу.

Приклад відображення інформації про контейнер, який знаходиться у вибраній точці вивозу зображено на рисунку 4.3.



Борщагівська вулиця, Київ

Заповненість баку: 71.4%

Тип сміття: Загальний

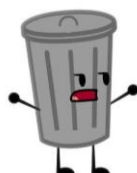


Рисунок 4.3 –Екран відображення інформації про контейнер

Натискаючи на кнопку вибору точок, для побудови маршруту, відкривається екран вибору точок. На екрані зображено поля, для вибору початкової та кінцевої точки, поле для вибору рівня заповненості смітників, через які буде будуватися маршрут.

Екран вибору точок, для побудови маршруту зображений на рисунку 4.4.

Початкова точка

Поточне місцезнаходження

Кінцева точка

Вірменська вулиця, Київ

Заповненість

90%

Тип сміття

Загальний

ПРОКЛАСТИ МАРШРУТ

Рисунок 4.4 – Екран вибору точок для побудови маршруту

За замовчуванням початкова точка вибирається як поточне місцезнаходження користувача. Вказуючи початкову та кінцеву точки, та натискаючи кнопку «Побудувати маршрут» мобільний застосунок починає визначати точки, через які потрібно побудувати маршрут та будує цей маршрут. Потім мобільний застосунок відкриває екран, який відображає користувачу побудований маршрут.

Натискаючи на поле для введення «Кінцева точка», відкривається екран з картою, на якій можна вибрати цю саму кнопку. Екран для вибору точки на карті зображений на рисунку 4.5.

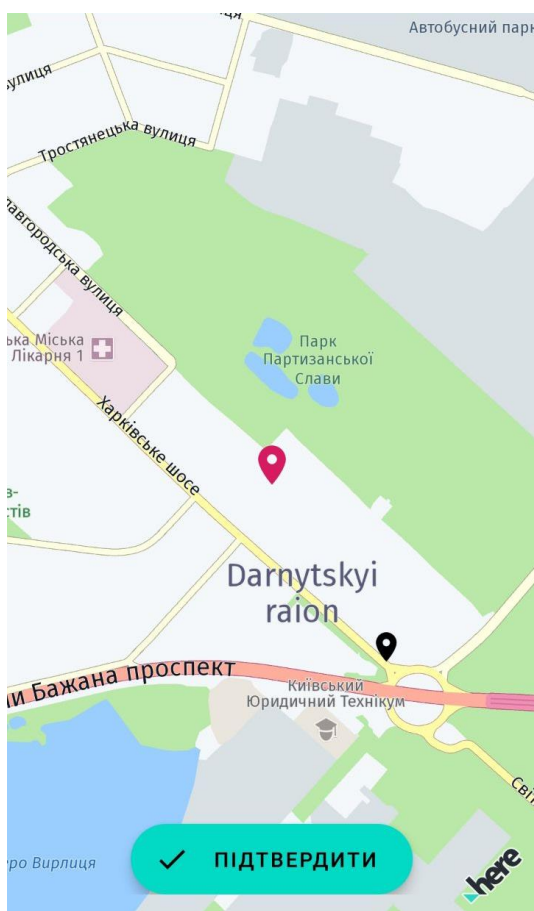


Рисунок 4.5 – Екран вибору точки на карті

Користувач має розмістити червоний маркер у заданому місці на карті та натиснути кнопку «Підтвердити». Після натискання на кнопку мобільний застосунок знову відкриває екран вибору точок.

На рисунку 4.6 зображений екран відображення побудованого маршруту.

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		57

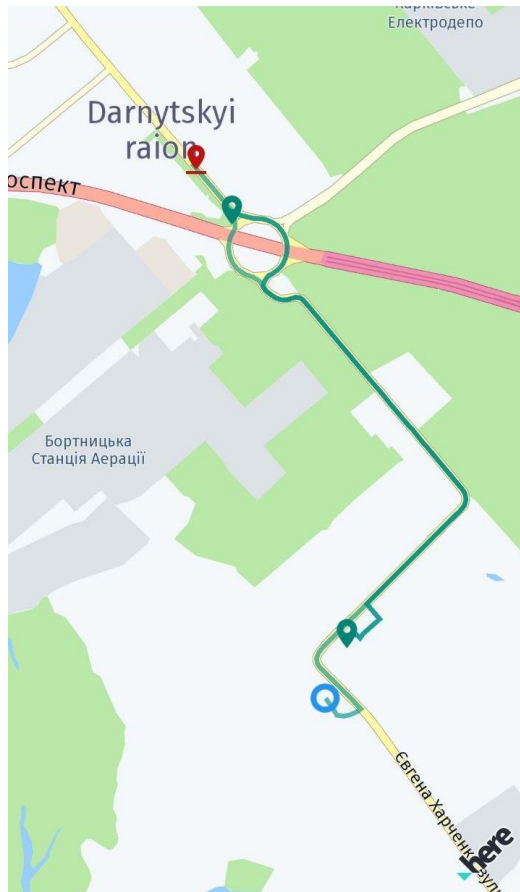


Рисунок 4.6 – Екран відображення побудованого маршруту

На побудованому маршруті відображаються поточне місцезнаходження користувача, точки вивозу із заповненими контейнерами та кінцева точка маршруту.

Висновки до розділу

Визначені системні вимоги до мобільного застосунку системи управління вивезенням відсортованого сміття.

Розроблено керівництво користувача мобільного застосунку системи управління вивезенням відсортованого сміття.

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		58

5 ДОСЛІДЖЕННЯ СИСТЕМИ УПРАВЛІННЯ ВИВЕЗЕННЯМ ВІДСОРТОВАНОГО СМІТТЯ

Проведено дослідження системи управління вивезенням відсортованого сміття на розробленому прототипі даної системи.

5.1 Мета дослідження системи управління вивезенням відсортованого сміття

Метою дослідження системи управління вивезенням відсортованого сміття є перевірка правильності отримуваних даних системою про заповненість контейнерів зі сміттям та правильність побудови маршруту через заповнені смітники на основі цих даних. Датчик, що встановлений на контейнері, повинен відправляти інформацію про заповненість центральному обробнику даних, який повинен зберігати ці дані в базу даних та надавати їх мобільному застосунку.

5.2 Огляд досліджуваної системи управління вивезенням відсортованого сміття

Досліджувана система є прототипом системи управління вивезенням відсортованого сміття та зображена на кресленнику ІТ61.070БАК.004 Д4.

Прототип системи управління вивезенням відсортованого сміття складається з таких компонентів: контейнери зі сміттям, мікроконтролер Arduino ESP8266 IOT Elecrow, ультразвуковий датчик відстані HC-SR04, Wi-Fi модуль, сервіс Firebase, Firebase Realtime Database, сервіс Here та мобільний застосунок системи управління вивезенням відсортованого сміття на операційній системі Android.

Прототип системи управління вивезенням відсортованого сміття містить лише один реальний контейнер зі сміттям, на якому знаходиться мікроконтролер із датчиком виміру відстані, який зображено на рисунку 5.1.

					ІТ61.070БАК.004 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дат		



Рисунок 5.1 – Контейнер зі сміттям, на якому знаходиться мікроконтролер із датчиком виміру відстані

Оскільки в прототипі системи управління вивезенням відсортованого сміття координати місцезнаходження задаються записом в базі даних напряду через Firebase, то для проведення дослідження можна задати будь-які координати точки вивозу контейнера зі сміттям. Такий підхід дозволяє нам записати в базу даних дані про абстрактні контейнери, тобто зробити імітацію інших контейнерів системи, які ніби то знаходяться в точках вивозу. Дані про абстрактні контейнери після запису в базу даних не будуть змінюватися.

У даному прототипі системи управління вивезенням відсортованого сміття контейнер зі сміттям із мікроконтролером та датчиком вимірювання відстані має такі координати: широта – 50.408423, довгота – 30.673048. Абстрактний контейнер 1 має такі координати: 50.393709, довгота - 30.653473. Абстрактний контейнер 2 має такі координати: 50.422147, довгота - 30.646516.

Мікроконтролер Arduino ESP8266 IOT Elecrow, який розташований на контейнері зі сміттям, підключений до мережі інтернет через Wi-Fi модуль. Принцип роботи мікроконтролера та датчика описано в 2.3.

У якості центрального обробника даних виступають сервіси Firebase, які описані в 2.4.1. Одним із сервісів є Firebase Realtime Database, який використовується у якості бази даних. Структура бази даних описана в 2.5.

5.3 Дослідження побудови маршрутів через точки вивозу заповнених контейнерів зі сміттям

Використовуючи прототип системи управління вивезенням відсортованого сміття проведено дослідження побудови маршрутів через точки вивозу заповнених контейнерів. Дослідження проводилось для контейнерів із загальним типом сміття, оскільки для інших типів сміття побудова маршруту проходить так само.

Заповненість контейнера зі сміттям на якому розташований мікроконтролер та датчик зображена на рисунку 5.2.

Вірменська вулиця, Київ

Заповненість баку: 83.9%

Тип сміття: Загальний

Рисунок 5.2 – Заповненість контейнера з мікроконтролером та датчиком

Контейнер зі сміттям на якому розташований мікроконтролер заповнений на 83.9% та знаходиться в точці вивозу на вулиці Вірменській в місті Києві.

Заповненість абстрактного контейнера 1 зображено на рисунку 5.3.

					ІТ61.070БАК.004 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дат		

Льва Ревуцького вулиця, Київ

Заповненість баку: 91.4%

Тип сміття: Загальний

Рисунок 5.3 – Заповненість абстрактного контейнера 1

Абстрактний контейнер 1 заповнений на 91.4% та знаходиться в точці вивозу на вулиці Льва Ревуцького в місті Києві.

Заповненість абстрактного контейнера 2 зображено на рисунку 5.4.

Горлівська вулиця, Київ

Заповненість баку: 92.7%

Тип сміття: Загальний

Рисунок 5.4 – Заповненість абстрактного контейнера 2

Абстрактний контейнер 2 заповнений на 92.7% та знаходиться в точці вивозу на вулиці Горлівській в місті Києві.

Місцезнаходження точок вивозу та місцезнаходження користувача мобільного застосунку зображені на карті на рисунку 5.5.

					IT61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		62

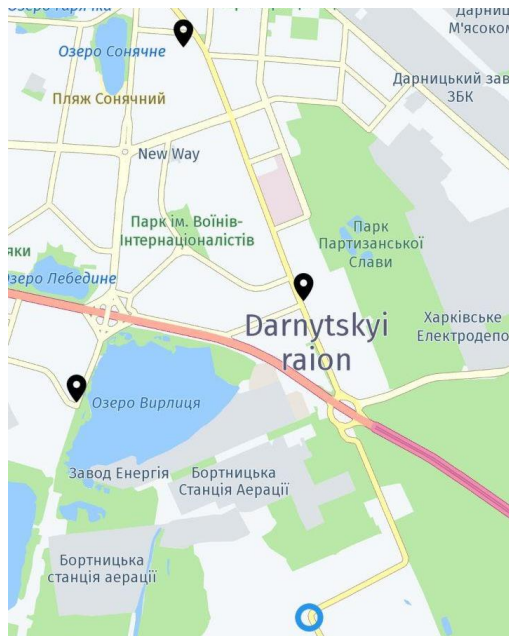


Рисунок 5.5 – Місцезнаходження точок вивозу та місцезнаходження користувача на карті

Початковою точкою маршруту вибране поточне місцезнаходження пристрою на якому працює мобільний застосунок. Кінцевою точкою вибрана вулиця Тепловозна. Заповненими визначено контейнери, які заповнені сміттям на 90% і більше. Тип сміття вибрано: загальний.

Побудований маршрут через точки вивозу із заповненими смітниками зображено на рисунку 5.6.

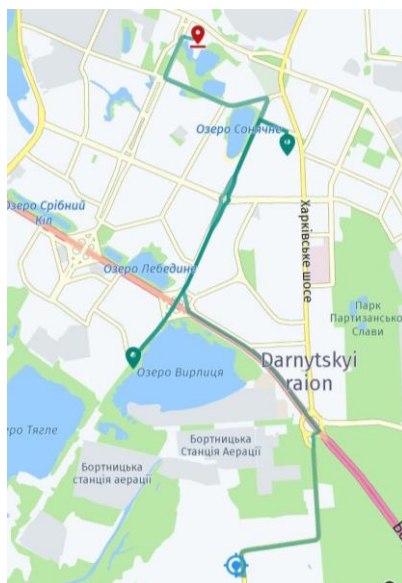


Рисунок 5.6 – Побудований маршрут через точки вивозу із заповненими контейнерами

Маршрут не проходить через точку вивозу, що розташована на вулиці Вірменській, оскільки контейнер там заповнений менше ніж на 90%. Додавши сміття в контейнер із датчиком та мікроконтролером, зміна заповненості контейнера показана на рисунку 5.7.

Вірменська вулиця, Київ

Заповненість баку: 90.7%

Тип сміття: Загальний

Рисунок 5.7 – Змінена заповненість контейнера з мікроконтролером та датчиком

Провівши побудову маршруту із тими самими параметрами, отримаємо побудований маршрут, що зображений на рисунку 5.8

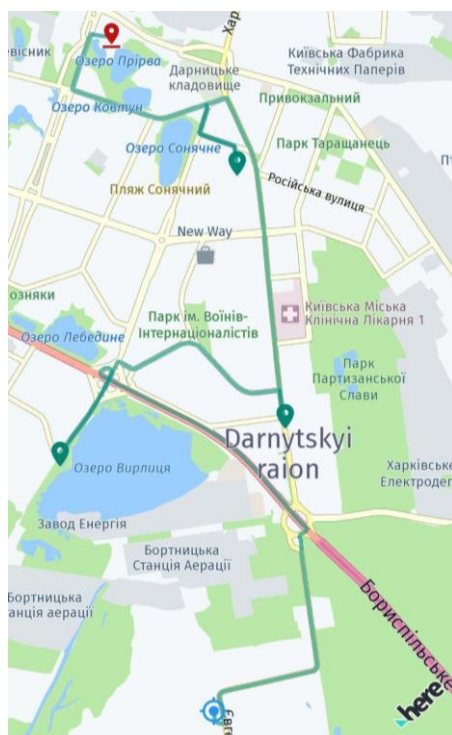


Рисунок 5.8 – Побудований маршрут через точки вивозу із заповненими контейнерами

					IT61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		64

Маршрут включає всі точки вивозу, які входять в систему управління вивезенням відсортованого сміття, так як всі контейнери в цих точках вивозу задовольняють умову заповненості контейнерів в 90% або більше.

Висновки до розділу

В результаті дослідження системи управління вивезенням відсортованого сміття на прототипі цієї системи, отримано підтвердження правильності побудови маршруту через точки вивозу із заповненими контейнерами зі сміттям. Датчик, що встановлений на контейнері, повинен відправляти інформацію про заповненість центральному обробнику даних, який зберігає ці дані в базу даних та надає їх мобільному застосунку. Мобільний застосунок системи управління вивезенням відсортованого сміття використовує ці дані для побудови маршрутів. Маршрут будується тільки через точки вивозу, які містять заповнені контейнери зі сміттям.

					ІТ61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		65

ВИСНОВКИ

Результатом даного дипломного проєкту є розроблена модель системи управління вивезенням відсортованого сміття та розроблений мобільний застосунок на операційній системі Android. Розроблений прототип системи управління вивезенням відсортованого сміття.

Отримана модель системи управління вивезенням відсортованого сміття оптимізує вивезення твердих побутових відходів, вирішуючи проблему вчасного вивозу твердих побутових відходів та зменшення собівартості вивозу сміття.

Розроблений мобільний застосунок на операційній системі Android дозволяє переглядати актуальні дані про заповненість контейнерів зі сміттям у точках вивозу та будувати маршрути лише через точки вивозу із заповненими контейнерами. Мобільний застосунок розроблений за допомогою мови програмування Kotlin та використовує сервіс Here для побудови маршрутів.

Проведене дослідження прототипу системи управління вивезенням відсортованого сміття, результати якого показують працездатність створеної моделі в цьому дипломному проєкті. Мобільний застосунок, який є частиною цієї системи управління вивезенням відсортованого сміття дозволяє досягнути поставленої мети в цьому дипломному проєкті.

					IT61.070БАК.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		66

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ролик А.И. Управление корпоративной ИТ-инфраструктурой / А.И. Ролик, С.Ф. Теленик, М.В. Ясочка – К.: Наукова думка, 2018. – 576 с.

2. За результатами дослідження GfK, за перше півріччя 2018 року кількість користувачів смартфонів у нашій країні досягла 85%, при цьому 50% українців шукали інформацію про бажаний продукт з мобільного телефону. URL: Режим доступу до ресурсу: <https://www.ukrinform.ua/rubric-economy/2510653-cogoric-50-ukrainciv-sukali-informaciu-pro-tovari-na-smartfoni-doslidzenna.html>

3. Мобильное приложение водителя URL: <https://play.google.com/store/apps/details?id=com.bigthree.asudriver>

4. Sensoneo Driver Navigation URL: <https://sensoneo.com/optimize-waste-collection-routes/>

5. RG-Soft Вывоз Мусора URL: https://play.google.com/store/apps/details?id=com.rg_soft.rgsm

6. Черемных С. В. Моделирование и анализ систем. IDEF-технологии : практикум / И. О Семенов, В. С. Ручкин – М.: Финансы и статистика, 2006 – 192с.

7. ESP8266 IOT Board URL: [https://www.elecrow.com/wiki/index.php?title=ESP8266_IOT_Board\(Arduino_IDE_or_NodeMCU_Lua_Programming\)](https://www.elecrow.com/wiki/index.php?title=ESP8266_IOT_Board(Arduino_IDE_or_NodeMCU_Lua_Programming))

8. Ультразвуковой датчик расстояния Ардуино HC-SR04 URL: <https://arduinomaster.ru/datchiki-arduino/ultrazvukovoj-dalnomer-hc-sr04/>

9. Firebase Documentation URL: <https://firebase.google.com/docs/android/setup>

10. Фаулер М. UML. Основы.: пер. с англ. / Скотт К. – СПб.: Символ-Плюс, 2002. – 192 с.

11. Android Architecture Patterns Part 3: Model-View-ViewModel URL: <https://medium.com/upday-devs/android-architecture-patterns-part-3-model-view-viewmodel-e7eeee76b73b>

					IT61.070БАК.004 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дат		

12. Шаблон «Спостерігач» URL: <https://refactoring.guru/uk/design-patterns/observer>
13. About Android URL: <https://developer.android.com/about>
14. About IOS URL: <https://developer.apple.com/ios/>
15. Cay S. Horstman. Core Java : книга : Prentise Hall, 2016. – 1038с.
16. Kotlin documentation URL: <https://kotlinlang.org/docs/reference/>
17. About Swift URL: <https://www.apple.com/ru/swift/>
18. About Objective-C URL:
<https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>
19. Google Maps Platform URL:
<https://developers.google.com/maps/documentation>
20. Here URL: https://developer.here.com/documentation/android-sdk-lite/4.3.3.0/dev_guide/topics/map-items-lite.html
21. OpenStreerMaps URL: <https://www.openstreetmap.org/about>
22. Android Studio URL: <https://developer.android.com/studio>
23. Thomas H. Cormen. Introduction to Algorithms : книга / Charles E. Leiserson, Ronald L. Rivest : The MIT Press, 2009. – 1328с.
24. Сервіси Google Play URL:
<https://play.google.com/store/apps/details?id=com.google.android.gms&hl=uk>

Додаток А
(довідковий)

Таблиці дипломного проєкту

Таблиця А.1 – Визначення рішень у сервісі Firebase

Назва компоненту сервісу	Призначення	Опис
Authentication	Аутентифікація користувачів	Надає сервіси та SDK для аутентифікації користувачі у розроблюваних застосунках.
Authentication		Підтримує аутентифікацію користувачів, використовуючи пароль, номер телефону та аутентифікацію через Google, Facebook, Twitter та інші.
Realtime Database	Збереження даних	Нереляційна cloud-based база даних. Дані в ній зберігаються у вигляді JSON документу, та синхронізуються в реальному часі з кожним клієнтом, який під'єднаний до неї, тобто всі клієнти одразу отримують актуальну інформацію, після того як вони були змінені в базі даних.
Cloud Firestore	Збереження даних	Гнучка нереляційна cloud-based база даних для розробки проєкт та мобільних застосунків. Дані зберігаються у вигляді моделі документа. Так само, як і Realtime Database, синхронізує дані з кожним клієнтом в реальному часі. Перевагою

Назва компоненту сервісу	Призначення	Опис
		є зручна та більш швидка робота зі складними моделями даних.
Cloud Storage	Збереження медіа-файлів	Потужне сховище для медіа-файлів, яке дозволяє швидко та безпечно завантажувати та скачувати медіа-файли, такі як: картинки, відео, аудіо, тощо.